THIS PAGE BLANK (USPTO)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 805 460 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
05.11.1997 Bulletin 1997/45

(51) Int Cl.[6]: **G11C 29/00**, G06F 11/27, G01R 31/3187

(21) Application number: **97302975.4**

(22) Date of filing: **30.04.1997**

(84) Designated Contracting States:
DE FR GB IT NL

(30) Priority: **30.04.1996 US 16516**

(71) Applicant: **TEXAS INSTRUMENTS INCORPORATED**
Dallas Texas 75265 (US)

(72) Inventors:
- **Hii, Kuong Hua**
  Singapore 809690 (SG)
- **Powell, Theo J.**
  Dallas, Texas 75428 (US)
- **Cline, Danny R.**
  Dallas, Texas 75904 (US)

(74) Representative:
Legg, Cyrus James Grahame et al
ABEL & IMRAY,
Northumberland House,
303-306 High Holborn
London WC1V 7LH (GB)

(54) **Integrated circuit having a built-in selft-test arrangement**

(57) An integrated circuit has a built-in self-test (BIST) arrangement (60). The built-in self-test arrangement includes a read only memory (ROM), (140) that stores test algorithm instructions. A ROM logic circuit (410) receives an instruction read from the read only memory and produces a group of output signals dependent upon the instruction. A BIST register 420 receives and stores the group of output signals from the logic circuit for controlling self-test of the integrated circuit

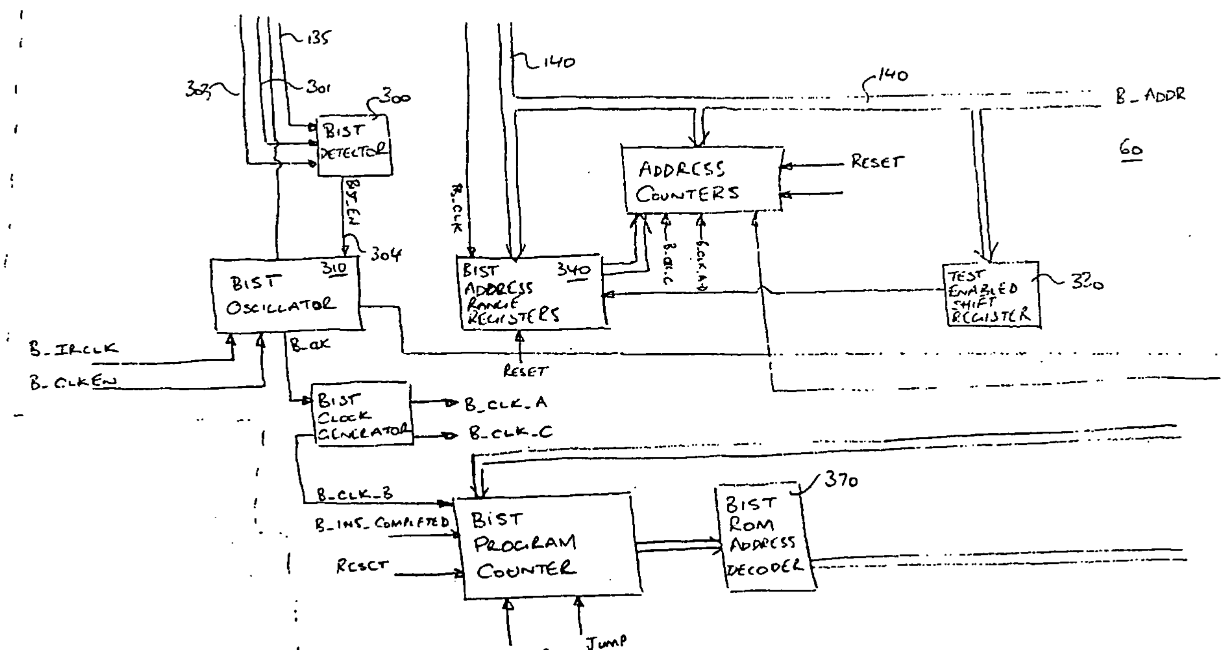


Fig 3

EP 0 805 460 A1

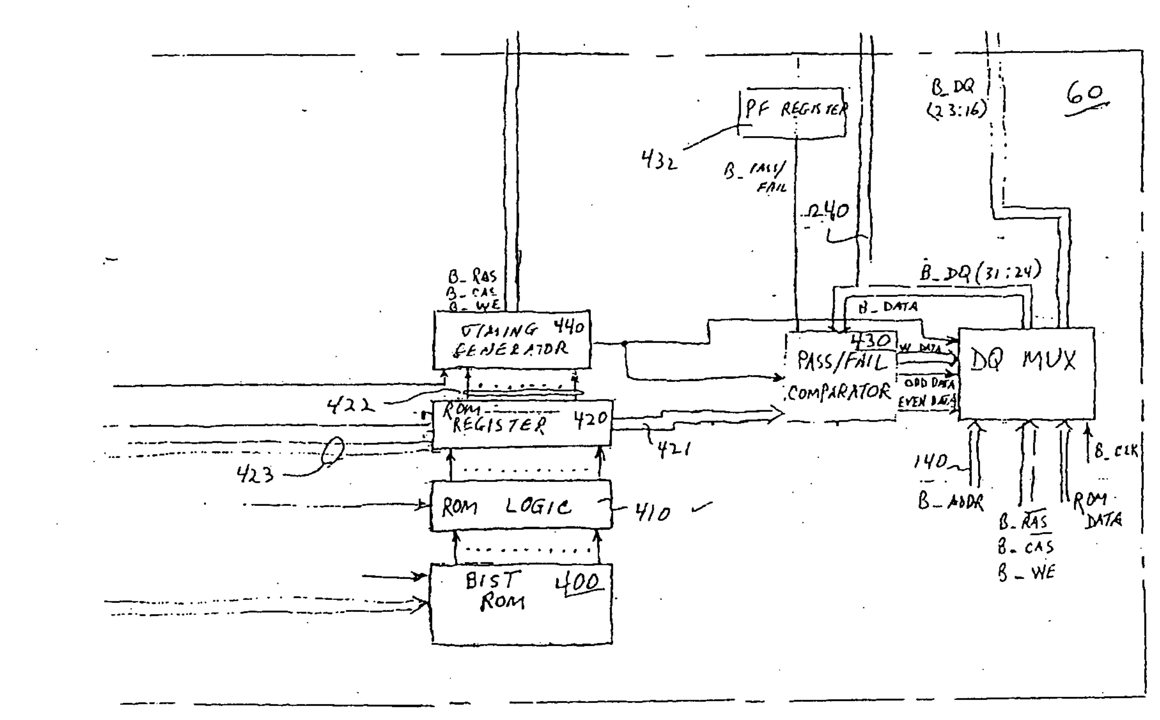

60
PF REGISTER
432
B_PASS/FAIL
240
B_DQ (23:16)
B_DQ(31:24)
B_DATA
B_RAS
B_CAS
B_WE
TIMING GENERATOR 440
430
PASS/FAIL COMPARATOR
DQ MUX
ODD DATA
EVEN DATA
422
ROM REGISTER 420
421
423
ROM LOGIC
410
BIST ROM 400
140
B_ADDR
B_RAS
B_CAS
B_WE
ROM DATA
B_CLK

FIG. 4

Description

FIELD OF THE INVENTION

This invention relates to the field of integrated circuit memory devices, and more particularly to a built-in self-test (BIST) arrangement for an integrated circuit memory device.

BACKGROUND OF THE INVENTION

In conventional testing of memory devices, a tester is used to supply the control signals such as RAS, CAS and WE, address signals, such as Ao-An, and data to the device under test. Outputs from the device under test are sampled by the tester to determine whether the device passes or fails. Testing of memories requires longer test times, as device density increases. This results in escalating test cost. As the capacity of integrated circuit memory devices increases to 256 Mbits and above, testing time per device becomes a major component of cost of integrated circuit memory devices.

One way to test integrated circuit memory devices in less time per device is to apply a single test data bit to several cells concurrently by multiplexing the single bit to the several cells in parallel. Some failures, however, cannot be screened unless a single cell is accessed at a time. With limited parallelism, i.e., a number of units being tested simultaneously, high test time also translates into a long manufacturing cycle time. Testing of one batch of memory devices requires most of the other devices to be waiting in queue to be tested while some of the memory devices are actually undergoing functional test. One solution would be to get more testers, but this is not practical as it involves even higher cost. The time to deliver a batch of tested memory devices to a customer increases as a result. Another solution is to apply the test from the testers in parallel to the devices under test. The problem with this solution is that the parallel leads occasionally cause good devices to fail because of cross talk among the parallel leads.

Thus there is a problem in finding some way to efficiently test large capacity memory devices without requiring an enormous amount of time on a test per memory device.

SUMMARY OF THE INVENTION

This problem and others are resolved by an integrated circuit which has a built-in self-test (BIST) arrangement. The built-in self-test arrangement includes a read only memory that stores test algorithm instructions. A logic circuit, receives an instruction read from the read only memory and produces a group of output signals dependent upon the instruction. A BIST register receives and stores the group of output signals from the logic circuit for controlling self-test of the integrated circuit. A pass/fail comparator circuit compares expected data bits with data bits written into and read from a memory array to determine whether the integrate circuit passes or fails the test.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be further described, by way of example, with reference to the accompanying drawings in which:

FIGS. 1-4 present a block diagram of an integrated circuit memory device arranged with built-in self-test circuitry in accordance with the present invention;

FIG. 5 shows a preferred arrangement of the integrated circuit memory devices of FIGS. 1-4;

FIG. 6 shows a plurality of devices mounted for concurrent testing;

FIG. 7 is a logic schematic of a test enabled shift register for BIST operation;

FIG. 8 is a block diagram of an address control circuit for the BIST arrangement;

FIGS. 9-12 are logic schematics used in the address control circuit of FIG. 8;

FIGS. 13A-13C are schematic diagrams showing the operation of the address buffer of the memory device;

FIG. 14 is a logic diagram of a BIST detector circuit;

FIG. 15 is a timing diagram showing operating conditions of the detector circuit of FIG. 14;

FIGS. 16-21 are logic schematics of BIST address register circuits;

FIG. 22 is a logic schematic of a BIST clock generator circuit;

FIG. 23 is a table showing the register positions for storing data representing what test operations should be enabled to run or not enabled to run for a specific self-test;

FIGS. 24 and 25 show a logic schematic of a BIST program control circuit that includes a program counter;

FIG. 26 is a logic schematic of the BIST ROM decoder;

FIG. 27 is a schematic diagram of a BIST ROM for storing test algorithm instructions;

FIG. 28 is a table of data stored in the BIST ROM for determining the operating sequences for several DRAM tests that can be run;

FIG. 29 is a table showing functions of bit stored in separate columns of the BIST ROM;

FIGS. 30 and 31 are logic schematics for an exemplary ROM logic circuit;

FIG. 31 is a logic schematic for a ROM logic circuit;

FIG. 32 is table of VHDL language description of a Pass/Fail comparator;

FIG. 33 is a truth table and definitions;

FIG. 33a shows a schematic of the BIST Internal Refresh Bit;

FIG. 34 shows a schematic of the Internal Refresh Oscillator;

FIG. 35 shows a schematic of the Address Reset Circuitry Sheetl;

FIG. 36 shows a schematic of the Address Reset Circuitry Sheet2;

FIG. 37 shows a schematic of the Column Counter;

FIG. 38 shows a schematic of the 10 Bit Counter "A";

FIG. 39 shows a schematic of the 10 Bit Counter "8";

FIG. 40 shows a schematic of the Control ROM;

FIG. 41 shows a schematic of a Data Multiplexer (MUX);

FIG. 42 shows a schematic of the Data Multiplexer (MUX);

FIG. 43 shows a schematic of Compare Circuitry;

FIG. 44 shows a schematic of Pass Fail detection circuitry;

FIG. 45 shows a schematic of the Refresh Counter;

FIG. 46 shows a schematic of the Row Counter;

FIG. 47 shows a schematic of the Timing Generator Sheet 1;

FIG. 48 shows a schematic of the Timing Generator Sheet 2;

FIG. 49 shows a schematic of the Timing Generator Sheet 3;

FIG. 50 shows a schematic of the Timing Generator Sheet 4;

FIG. 51 shows a schematic of the full BIST logic;

FIG. 52 shows a schematic of the full BIST functional logic;

FIG. 53 shows a schematic of the ROM Driver logic;

FIG. 54 shows a schematic of the Address Reset Enable;

FIG. 55 shows a schematic of the Row Counter;

FIG. 56 shows a schematic of the Column Counter;

FIG. 57 shows a schematic of the Refresh Counter;

FIG. 58 shows a schematic of the 10 bit Counter "A";

FIG. 59 shows a schematic of the 10 bit Counter "B";

FIG. 60 shows a diagram of the BIST block and Interface;

FIG. 61 shows a diagram of the BIST layout location;

FIG. 62 shows a diagram Algorithms and Timing sets;

FIG. 63 shows a diagram the entry/exit state transitions;

FIG. 64 shows a timing diagram;

FIG. 65 shows a diagram of the pin out and key signals for BIST;

FIG. 66 shows a table for clock multiplexing;

FIG. 67 shows a diagram the ROM Memory Map;

FIG. 68 shows a timing diagram for array access and program control;

FIG. 69 shows a timing diagram;

FIG. 70 shows a block diagram of the ROM interface;

FIG. 71 shows a diagram of the Instruction Timing;

FIG. 72 shows a diagram of the control words;

FIG. 73 shows a table of the timing sets;

FIG. 74 shows a diagram of the Timing Generator;

FIG. 75 shows a diagram of the Timing Sets;

FIG. 76 shows a diagram of the Multiplexer;

FIG. 77 shows a diagram of the DUT (Device Under Test) Timing;

FIG. 78 shows a diagram of the Address Control Bits for entry and test control;

FIG. 79 shows a diagram of the DQ (Data/Output) multiplexing;

FIG. 80 shows a diagram of the address range selection;

FIG. 81 shows a diagram of the ROM programming schematic, option low;

FIG. 82 shows a diagram of the ROM programming schematic, option high; and

FIG. 83 shows a diagram of the Multiplexing Circuits.

DETAILED DESCRIPTION

Referring now to FIGS. 1-4, there is shown the block diagram of an integrated circuit dynamic random access memory device 50 including a built-in self-test (BIST) arrangement 60. The device 50 is designed to operate as a synchronous random access memory during normal operation. Alternatively the built-in self-test arrangement operates in a distinctive self-test mode at times while the device 50 is not operating in the normal mode. The built-in self-test arrangement is designed such that all the test signals are generated internally to a device, and the arrangement only takes a simple setup to get the device into a self-test mode to perform a self test. With the simple setup up, the built-in self-test arrangement performs a memory self-test in a cost effective procedure. The arrangement also allows many devices to be tested in parallel without being limited by tester resources.

While the device 50 operates in the normal mode, it operates like a well-known synchronous dynamic random access memory device. Row and column addresses produced by a digital processor, such as a microprocessor, are time multiplexed by way of the address bus A0-A13 into an address buffer 100 until control signals RAS and CAS , respectively, which load them. Thereafter they are decoded either by the row decoder 200 or the column decoder 210. The control signals RAS and CAS also are produced by the digital processor, which is not shown.

Depending upon the state of the write enable signal WE_from the digital processor, data is either written into or read out from storage cells located in the banks of the memory array 220. Data, to be written into the banks of the memory array 220, is applied to the device 50 by way of a data bus DQ0-DQ31. This incoming data is buffered in a data buffer circuit 230 and forwarded by way of an internal data bus 240 to the memory array 220 where it is stored until it is written over or until it is no longer refreshed at specified intervals. While data is stored in the memory array 220, that data can be addressed and read out of the array 220 through the internal data bus 240 and the data buffer 230 to the data bus DQ0-DQ31. This data bus DQ0-DQ31 typically is connected to the data receiving and sending terminals of a digital processor, such as a microprocessor that is not shown.

Because the memory device 50 is a synchronous dynamic random access memory, a system clock signal CLK is applied to the device from an external source for synchronizing its operation with the digital processor, peripheral devices, and control circuitry that are connected in a system. The system clock is the clock which controls operation of the digital processor. The clock signal CLK is applied to a clock buffer 110, which outputs an internal clock signal CLK for operating the memory array 220 during normal operation. This internal clock signal CLK controls operation of address decoding, writing data to the memory array, and reading data out of the memory array during such normal synchronous memory operations.

The control signals RAS , CAS , and WE , which are applied to the memory device 50 from the digital processor, are applied to a control signal buffer 120. During normal mode operation, these control signals pass through the control buffer 120 and a control bus 125 to the main control unit 130 of the memory array 220. At the memory array 220 during normal operation, these control signals RAS , CAS , and WE together with the internal system clock signal CLK control operations of the array 220, as previously described.

Normal mode operation and self-test mode operation are two separate and distinct operations of the memory device 50. Those two modes occur alternatively. Thus while the device 50 operates in its normal mode it is not able to inadvertently go into its self-test mode. Also while it is in its active self-test mode, it cannot inadvertently go into its normal mode. These are conditions that are imposed upon the operation of the device 50 by the built-in self-test arrangement 60 to be described.

As just mentioned, the self-test mode of operation is different from the normal mode of operation. Self-test mode is entered only upon power up of the memory device 50. Special signal conditions are applied at that time to put the device 50 into the self-test mode.

In this proposed BIST scheme, only DC signals are needed external to the device to enter the self-test mode and

to actually proceed through the self-test. The following types of external DC signals are supplied:

a) An overvoltage on one of the multiplexed address pins (e.g., pin A4.)
b) A switch on CS_ which allows both a '0' and a '1' to be connected to it.
c) A logic level of '0' or a '1' on the rest of the address pins for tests and options selections.
d) Pass, fail detection on one of the output pins.
e) An output detector pin to identify completion of the self-test operations.

With the above set up, only DC signals are involved. No complex timing is needed. Thus there is no need to be concerned with signal quality. The signal quality to the device is always good since the signals are DC. Multiple devices can be put on a BIST board and self tested simultaneously. Test cycle time can be reduced per device since BIST can be applied to all devices on the BIST board in parallel.

FIG. 6 shows an arrangement for mounting several integrated circuit devices which are interconnected in parallel so that they all can be set up at once for self-test. Once set up the self-test can be run concurrently on some of or all of the devices.

While in self test operations, BIST circuits take over control of all the signals such as CLK, RAS_, CAS_, WE_, the address bits, and the data bits. For example, a BIST address bit B_Ax is generated by the BIST circuits. During a self-test operation, the BIST address bits replace the externally generated address bits Ax. The BIST address bits B_Ax interface with the main circuit right out of the address buffer.

During normal mode operation, the BIST address bits B_Ax are not generated and are ignored. During power up, if a BIST request is detected, the address lines are used to bring in information into the BIST arrangement. Referring now to FIG. 7, a test enabled shift register 330 receives and stores data that determines whether or not specific tests are enabled. The data are stored in the shift register at the right-hand side of FIG. 3. During active self-test mode, BIST generated address signals are used for operating the device and externally applied address signals Ax are ignored.

Referring now to FIG. 8 there is a block diagram of a BIST address control circuit 65 that controls generation of array addresses during an active self-test operation. FIGS. 9, 10, 11 and 12 are logic schematics of circuits included within the address control circuit 65 of FIG. 8.

The BIST signals interface with the main circuit as close to the buffer as possible to ensure that they closely simulate conventional test conditions. This will allow the BIST circuits to test as much of the memory device circuitry as possible.

FIGS. 13A_13C show the interface between BIST generated signals and the main devices. The main function of the BIST generated signals is to replace the external signals that the memory device receives. Thus at the BIST interface there is a multiplex circuit which chooses between the BIST generated signals B-Ax and the external signals Ax from outside. The address lines are being used for multiple purposes, besides the above two cases, they are also used in power up to bring information into the BIST circuit, as illustrated in Figure 13B. Control signals B_PADDIS and B_IN_EN, generated by the self-test arrangement control how the multiplexer operates at anytime.

The address bus A0-A13 of FIG. 1 is used for applying the special signal conditions for the self-test mode during and after power up.. An overvoltage signal is applied by way of one lead of the address bus A0-A13 to a BIST detector circuit 300, which responds to the overvoltage condition by putting itself in a standby self-test condition. In this description the address lead A4 is used as an exemplary over voltage lead. While the BIST detector circuit 300 remains in its standby condition, it allows the built-in self-test arrangement to prepare for testing by accumulating information about a specific test to be run. Typically the specific test will be selected from a large group of tests which might be run. The built-in self-test arrangement will remain in the self-test standby condition until another input signal is applied. Meanwhile data can be written into or read out of the memory array 220, as if it were in the normal mode because the built-in self-test arrangement 60 is in standby rather than in active self-test mode.

The self-test arrangement 60 is put into the active self-test mode from standby by applying a high level signal CS_ by way of a lead 135 to the BIST detector circuit 300. The built-in self-test arrangement 60 and the memory device 50 will operate in the active self-test mode as long as the signal CS_ remains at the high level and then return to normal operation when the signal CS_ goes to its low level.

Referring now to FIG. 14, the BIST detector circuit 300 includes input terminals 301, 135, and 303 for receiving, respectively, a signal VUPB, an over voltage signal A4 from the lead of the address bus A0-A13, and the control signal CS_. An output lead 304 carries a built-in self-test enable signal BIST EN that indicates when the device 50 is operating in its self-test mode. The BIST detector circuit 300 detects a BIST request during power up only. To detect a BIST request requires an overvoltage to be detected at power up and the signal CS_ having a value of '0' at that time. When the circuit 300 detects these two conditions at power up, the chip goes into a BIST standby mode. While the circuit 300 is in BIST standby mode and the signal CS_ goes high, the circuit 300 enters the self-test active mode.

FIG. 15 presents the timing diagram for the operation of the BIST detector circuit 300 of FIG. 14. Several key signals are presented in FIG. 15 together with their interdependency. It is noted that all of the signals applied to the

BIST detector circuit 300 are fundamentally voltage levels. As will be demonstrated subsequently herein, no external fluctuating control signals need to be applied to the memory device 50 during the active self-test mode. All signals for performing the repertoire of tests are produced by the built-in self-test arrangement 60 on the device 50.

A more detailed description of the arrangement and operation of the BIST detector circuit 300 is presented in a co-pending patent application, Serial No. . (TI-22640) which was filed concurrently herewith. In that patent application, the BIST detector circuit 300 is referred to as an entry and exit control circuit. The subject matter of that patent application is incorporated herein by reference thereto.

Two signals, A4 and control signal CS_ are multiplexed for built-in self-test entry. The signal A4, or any other address bit, is multiplexed as an overvoltage. Control signal CS_ is used for timing BIST entry and EXIT. A BIST request is detected if an overvoltage is detected and the control signal CS_ is low when the device is being powered up. If an overvoltage is detected at the falling edge of VUPB, the device will go into BIST standby mode. The device will only go into BIST active mode if the control signal CS_ is subsequently pulled high. This will allow full control of the time to enter the BIST active mode. During BIST operation, if CS_ is pulled low, the device will exit BIST immediately. Normal operation occurs when CS_ is low. This prevents the BIST active mode from occurring during normal mode operation.

In BIST standby mode, the device can operate as if it were in normal mode. Note that the control signal CS_ behaves normally while the device is operating in normal mode. It is only in the BIST mode that the control signal CS_ function differently. Design for test (DFT) modes can be executed when device is in BIST standby mode. One of the ways to use DFT mode while in BIST standby is to have the address range mode load start and stop addresses before the BIST active mode operation.

Referring now to FIGS. 16-21, there are address range registers for storing address information used during self-test operations. When a BIST request is detected, an internal signal B_SR_LOAD is generated to load in the information located on the address lines except the address for the overvoltage signal on lead A4. The information loaded in contains two sets of information. Test selection data on the address leads A0-A3, A5-A10 determine whether or not each specific test is to be run in BIST active mode. Secondly, BIST operation options such as internal external clock option, full/sub array option, enable disable output option.

In FIG. 3, there is shown a BIST oscillator circuit 310 that includes an arrangement for generating a continuously repetitive clock signal B_CLK which is used to produce a group of specific clock signals for controlling different parts of the built-in self-test operation.

FIG. 22 is a logic schematic of a BIST clock generator circuit 350 of FIG. 3. It derives BIST clock signals B_CLK_A, B_CLK_B, and B_CLK_C.

Upon entering the active self-test mode, the clock signal B_CLK is applied to the clock buffer circuit 110 of FIG. 1 for taking over control of the clock buffer circuit 110, which thereafter during the active self-test mode produces the clock signal B CLK to control access to and the operations of the memory array 220, as well as parts of the built-in self-test arrangement 60. Initially data from the address buffer 100 is transferred by way of an internal address bus 140 to an enabled tests circuit 330 in FIG. 3. Exemplary circuit 330 is a shift register arrangement that stores data identifying a specific test, or a group of tests to be run during the relevant active self-test mode. This data may be, for example, a high level for each test to be run and a low level for each test that is not to be run. Once the selected test data is stored in the enabled tests circuit 330, a group of addresses may be stored in the address range registers 340.

Referring now to FIG.23, there is shown a diagram of the information that is stored in the enabled test circuit 330. In FIG.23, the address bits positions A0-A3, respectively, represent the gross test, the pause test, X march and Y march. Address bit positions A5-A10, respectively, represent short disturb, long disturb, page disturb, burn in, write one row, and read one column. Address bit position All and bank addresses BA0 and BA1, respectively, represent output enable, sub array option, and internal clock/external clock selection.

A clock is needed for the BIST operation. BIST circuit has an internal oscillator which provides this clock with a fixed frequency. There is an external clock option built in for engineering and debug purposes which allows the clock frequency to be varied if needed. The decision to use an external or internal clock is loaded in when a BIST request is detected during power up.

The previously mentioned address range registers 340 include a group of four registers which receive and store, respectively, row and column addresses which determine start and stop addresses in the memory array 220 where the test or tests should commence and end. Such addresses are used at the discretion of the person responsible for testing the device 50. A default condition runs the tests throughout the entire memory starting at row address 0 and column address 0. Thus the address range registers 340 are reset to zero when the device 50 is initially put into the active self-test mode. Thereafter the range addresses may be stored if desired during a DFT mode load operation.

During prototype debug on any early production chip, it is unlikely that a complete array will work correctly. Under such circumstances, the BIST test will always fail since there will always be some cells which do not work and will cause failure of the BIST tests. This means that the BIST logic cannot be completely checked out since there will never be an application of BIST which permits the BIST logic to return a "PASS" signal until a completely good part is manufactured. A second problem occurs during debug of the chip. If the BIST is limited to the complete array test, it cannot

be used to target a subset of the array as an aid for debug.

Sub array testing permits the starting address to be any location and the ending address to be any location and they can be loaded into the BIST address range registers via a DFT mode. The BIST testing is applied between the starting and ending locations including one address location of the starting address is the same as the ending location. The address ranges can be changed each time prior to BIST application thereby permitting the BIST to be applied to islands to fault free areas through repeated testing.

Referring now to FIGS. 24 and 25, there is shown a program control circuit that includes a program counter 360. The BIST program counter 360 is included in the built-in self-test arrangement 60 for controlling test sequences. Upon initiation of the active self-test mode, the BIST program counter 360 is reset by a signal B RESET to its zero state. This is a sequential logic arrangement in which its current state, in combination with the state of data furnished by a BIST ROM register determines the next state of the BIST program counter 360. The state changes in response to a clock signal B CLK B applied from the clock generator 350.

FIG. 26 is a logic schematic for a BIST ROM address decoder 370, shown in FIG.3. The state of the BIST program counter 360 is represented by a group of binary signals that are applied by way of a bus 361 to the input of the BIST ROM address decoder 370. The group of input binary signals are decoded into a 1 out of 64 code for selecting a row of data from the BIST ROM 400.

FIG. 27 shows the schematic layout of an sixty-four word by twelve bit BIST ROM 400 of FIG.4. The BIST ROM 400 is a sixty-four row read only memory that stores sequences of instructions for controlling several different test routines used to determine the operability of the memory array 220. The proposed BIST scheme has ten algorithms stored in a ROM. Each algorithm is typically made up of a series of instructions. The ten algorithms take up sixty-four ROM words and each ROM word has twelve bits. Each row address applied to the BIST ROM 400, accesses a row of data stored therein in response to a clock signal B CLK.

The first instruction in an algorithm is an instruction to determine if that test is being enabled. Whether the test is enabled or not is decided at power up when test selection information is loaded into the test enabled register. All or any subset of tests can be selected. A block of test code is skipped if a '0' logical value is loaded in the corresponding tests enabled register.

For most algorithms, the last instruction in an algorithm tests for 'inverted pattern'. In a typical test, two data patterns ('0' and '1') need to be performed. This means each test is executed twice, once for each pattern. This instruction looks at a register to determine if the current test is executing the normal pattern (pattern '0'). If it is, then the program counter will jump to the start of the test and repeat the test with an inverted pattern. If the instruction determines that it is executing an inverted pattern, it will simple increment the program counter by 1 and move on to the next test since both data patterns have already been executed.

Referring now to FIG.28, there is shown the table of data stored in the BIST ROM 400. In the left-hand column is the list of names of ten algorithms which represent the tests which may be selected to be run plus an instruction that all tests have been completed.

The first algorithm GROSS is for running a gross test. There are four rows of data, each representing one instruction for the gross test. Addresses for the instructions are shown in hexadecimal code in the second column from the left. The right-hand column presents the mnemonic name for each instruction. The main block of the table presents the data which is stored in the BIST ROM 400. There are twelve columns of data in the table. In the table, there is a bold horizontal line setting off the beginning and ending addresses of each algorithm. Thus there is a bold line below the address hex 3 which is the fourth instruction in the algorithm GROSS.

The four instructions in the algorithm GROSS are jump not test enable (jnte) to pause, write all cells zero, read all cells with expected data zero, and invert data and jump if not previously inverted (divnj). The Z1 is the label at which to jump.

Referring now to FIG. 29, there are two major types of instructions. The first type is the program control instruction. This type of instruction deals with the flow of the program. The program control instruction are introduced to control BIST operation. The second type is the array access instruction which control how the cells of the array are to be accessed and written to and/or read from. They are basically the same type of instruction usually found on a tester which are translated into BIST ROM format. An instruction that reads back a pattern from the array (220) is an example of a array access instruction.

The instruction is divided into two parts. The first six bits of an instruction defines the actions to be taken and the last six bits is the data associated with the instruction.

For an array access instruction such as read whole array, the first six bits define the type of array access, whether it is a write or a read or both, whether the full array, only the rows, or the columns are accessed. To read from the whole array, Read (bitll), X(bit9) and Y(bit8) are set to 1. The last six bits of the instruction provides information on how the whole array is read including the timing sets to be used (Tset0, Tset1), the data values (ED) and pattern () and whether the address is to be incremented or decremented.

For a program control instruction such as the 'test enable' instruction. The four most significant bits are '0', Bit7

and Bit6 determines the program control type. The last six bits provides the address to be jumped to if indeed the decision is made to jump.

There is an unconditional jump instruction. '110011' for the most significant 6 bits, and the address to jump to for the 6 least significant bits.

The last instruction in the ROM, see FIG 28, is an idle instruction to signal the end of BIST operation. The last six bits of this instruction holds the revision number of the current 256M.

There are many possible combinations of instructions that can be programmed with the current circuits. If a new algorithm is needed for a BIST operation. It can be included by simply reprogramming the ROM. Combinations of options available to make up an instruction are as follows:

Timing sets	Access mode	Addressing	Array size	'0' or '1'	Data Pattern
TSETA	READ	INC	FULLA	PATTERN0	CKBD
TSETB	WRITE	DEC	ROW	PATTERN1	ALTERNATE
TSETC	RMW		COL		

Referring now to FIGS. 30 and 31, there is shown logic schematics for a ROM logic circuit 410. As each of the instructions is read out of the BIST ROM 400, the data is applied to the input of the ROM logic circuit 410, which is a combinational logic circuit that decodes the twelve bits of data of each instruction word. Output signals from the ROM logic circuit 410 are applied to the data inputs of a ROM register circuit 420 where the data are stored for the duration required to complete execution of the instruction.

When the built-in self-test arrangement 60 is put into the active self-test mode, the BIST program counter 360 is reset. This initial state of the program counter 360 is decoded through BIST ROM address decoder 370 to produce a row address signal for the BIST ROM 400. After the row address is applied and in response to a clock signal, the data from the selected row of the BIST ROM 400 is read out. All of the data read out from the selected row is applied to the input of the ROM logic circuit 410.

FIGS. 30 and 31 present an exemplary logic schematic diagram of the ROM logic circuit 410 that performs desired combinational logic functions on the row of data applied from the selected row of the BIST ROM 400. The circuit 410 produces a group of output signals resulting from logical processing through the circuit 410. This group of output signals from the circuit are applied in parallel to and are stored in the ROM register 420, which is arranged to forward them by way of a BIST data bus 421 to the pass/fail comparator circuit 430, by way of a group of leads 422 to the BIST timing generator 440, and by way of a program counter input bus 423 to the BIST program counter 360.

Referring now to FIG. 31, there is shown a logic schematic of an exemplary ROM register circuit 420. The data applied onto the program counter input bus 423 is accepted by the program counter 360 only when a program control instruction is being executed. Data applied to the BIST timing generator controls generation of self-test signals, such as B_RAS_, B_CAS_ and B_WE_, which perform the functions of their similarly named control signals RAS_, CAS_, and WE_, used by a microprocessor to access the memory during normal operation. Signals, applied by way of the BIST data bus 421 to the pass/fail comparator circuit 430, include memory access instructions and a data bit.

FIG. 44 presents the logic schematic diagram of an exemplary pass/fail circuit 430 that can be used in the built-in self-test arrangement 60. The pass/fail circuit 430 is responsive to control signals and a data bit received by way of the BIST data bus 421 from the ROM register 420 to produce a sequence of groups of data signals to be written into at least one bank of the memory array 220 in response to a write instruction from the BIST ROM 400. Pass/fail circuit 430 also receives control signals and a data bit by way of the BIST data bus 421 and read out data on a DQ bus 431 from the memory array 60, in response to a read instruction. In this instance, the data bit from the BIST data bus 421 is processed to agree with the state of a prior-existing data bit that was written into the array location/locations from which the data on the DQ bus 431 is read. The processed data bit is referred to as an expected data bit. This expected data bit is compared with the data read from the memory array 60 and the result of the comparison is a Pass signal if the compared data are equal and is a Fail signal if the compared data are not equal. The Pass signal, indicating that the circuits traversed by the written in and read out data and the storage cell are operating correctly, is a low level signal. A Fail signal, indicating that some part of the circuits traversed by the written in and read out data or the storage cell is malfunctioning, is a high level signal.

A Fail signal is transmitted by way of a lead B_Pass/Fail to a PF register 432 where it is stored. Subsequently the stored Fail signal can be conveyed through a DQ buffer circuit 230 to an external pad of the memory device. PF register 432 is reset upon entry of the active self-test mode. Pass signals are ignored and in effect discarded because they are a low level and the PF register is reset to begin the test operation. An assumption is made that the device is operable. So a single test failure is the only information of importance to be retained.

Referring to FIG. 32, a VHDL language description of the Pass/Fail comparator is shown. VHDL language is a

standard language for describing logic circuits used by designers today. Using signal CKBDI for checker board data, B_ALTERNATE for alternate data, B_RDATA for expected data (ED) and B_WDATA for write data (DAT), the data which is expected to write and to compare the read data is calculated. The read data is compared to the calculated expected data to determine if the test passed or failed.

Referring to FIG. 33, the functional table is given. The 32 bits of the array 220 compressed to 4 data bits using 8 DQ lines. The compression table is shown in FIG. 33. B_PF_ results gives the pass or fail depending on the corresponding values shown in the table.

Build In Self Test (BIST)

10.00 Overview of BIST

In memory functional testing, various algorithms are used to test a device. Tester are normally used to generate test signals, which represent the test algorithms, to the device. The outputs of the device are then sampled by the tester to determine pass or fail. For very high density memory devices such as 256M SDRAM, test time is expected to be very high thus making it not practical or cost efficient to do all functional testing on a tester. Build In Self Test (BIST) circuits are designed to replace the testers. The role of BIST circuits (Fig 51,52) is to generate all the test signals associated with the test algorithms internal to the device. Instead of using a tester, a device under test will power up in BIST mode, if certain voltage conditions are satisfied, and test itself. In BIST mode, BIST generated signals takes control of control signals such as RAS and CAS as well as all the addresses and data.

Advantage of BIST : Test Cost savings.

The build In self circuit is made up of the following (Fig.60):

Oscillator with a 50Mhz frequency
Overvoltage detector on A4
64 word X 12 Bit ROM
6 Bit Program counter
6 to 64 decoder
14 bit X register, 9 bit Y register and 14 bit refresh register
Timing generators
Address range counters
Shift register for storage of enabled test algorithms
Internal pattern data generator
Pass fail compare circuits
Output multiplexer.

BIST circuit takes up 1.9mm^2 in silicon area and occupies about 0.5% of the total chip area. It is located at the right most end of the chip, critical speed interface logic such as the address interface is placed near the speed path to avoid slowing down the normal operation. Control logic, ROM and various other BIST logic is placed on the end of the chip where silicon area is lower priority than in the center or intersection areas of the chip (Fig 61):

10 algorithms were implemented in the 256M BIST scheme, Summary for them are show (Fig 62)

Behavior for BIST circuits has been described in VHDL code and simulated using QVHDLsim. ALL BIST circuits except for the oscillator, over voltage detector and ROM were synthesized using Autologic 2.

10.01 BIST Entry and Exit

BIST Entry and Exit scheme is designed to satisfy the following requirements.

1. Simple Entry and Exit sequence with no timing required in order for BIST to operate with low cost BIST boards.
2. No inadvertent BIST entry.
3. Ability to alternate between Normal and BIST mode for read and write for testing BIST circuit operation.

It is important to have a simple entry and exit sequence so that BIST operation can be done on a simple BIST board with very high parallelism without having to worry about signal quality.

BIST Entry :

To do this, BIST entry is designed such that only DC signal is needed. To get into BIST, all that is needed is an overvoltage on A4 and low on CS during power up (and high or low on other pins depending the options chosen). This will put the device into a BIST standby mode (Fig 63).

While in standby mode the device will operate as if it is in normal mode and it can perform read, write and regular MRS or even DFT commands.

A rising edge on the CS pin will start the BIST operation while device is in BIST standby mode.

It is important to have a delay from overvoltage detection to actually starting BIST operation because it gives the user the opportunity to start the BIST operation at a time suitable to him.

It allows the user to use a DFT mode to set the address range, for example or giving the device enough time to settle down to a steady state before BIST operation began. It will also allow the user to write a row in normal mode and later enter BIST to read back the data in the array in BIST mode (or vice versa) to check BIST circuit functionality.

BIST Exit :

A falling edge of CS will exit all BIST operation unconditionally and the only way to get back into BIST mode is to power down and then power up again with overvoltage.

It is also imperative to prevent inadvertent BIST entry. To achieve this, overvoltage can only be detected at power up at the falling edge of VUPB. It cannot be detected at any other time. This and the CS interlock during power up gives the device only one chance to go into BIST standby mode. This minimizes the chance of inadvertent BIST entry during device operation.

When an overvoltage is detected at A4, it generates a B_SR_LOAD pulse. This pulse loads the information available on the otherl3 addresses into a shift register. These contain the information on which tests are to be executed while in BIST as well as information for clock option, array size option and output enable option. Details on these can be found in B_SHIFT block.

The Entry and Exit timing sequence for BIST operation is summarized here (Fig 64):

An example of a power up setup is on an example pinout (Fig 65).

10.1 BOV

INPUTS: B_OVDETECT, B_DONE, TLBADDRNG, PBCSB_BIST, VUPB

OUTPUTS: B_SR_LOAD, B_IN_EN, B_EXTCLK, B_PADDIS BIST_EN BIST_MCEN

OF CIRCUITS: 1/chip

LOCATION: right side of chip

The function of this circuit is to detect overvoltage during power up. It does not detect overvoltage at any other time.

During power up, at the falling edge of VUPB, a pulse VUPBN which gives about 16ns to charge up B_OVERDETECT is generated. The falling edge of VUPBN will set the overvoltage latch OVLATCHB to low if B_OVERDETECT is a high and if PBCSB_BIST is low. A low on OVLATCHB signals the detection of overvoltage and entry to BIST standby mode.

During the BIST standby mode, if PBCSB_BIST subsequently goes high, BIST_EN will go high to signal the start of BIST operation. A high to low transition of PBCSB_BIST will reset the overvoltage latch and set OVLATCHB to high . It also brings BIST_EN to low signaling BIST exit.

If there is no overvoltage detected during power up, B_OVERDETECT will be low and it will not get into BIST standby mode.

If overvoltage is detected at power up and OVLATCHB latch is set at power up, B_SR_LOAD pulse is generated to load the information on the address lines into a shift register. This information determines which test will be performed in BIST mode.

B_PADDIS and B_IN_EN are used to control the multiplexing of BIST signals and external signals. In normal operation, external signals are used. This is the normal mode in which the device operates in.

If overvoltage is detected at power up and the device is ready for BIST operation, B_IN_EN will be set to high to allow address signals to get to BIST circuits. In TLBADDRNG DFT mode, where the start and stop addresses are loaded into the BIST circuit, B_IN_EN is also high.

In BIST operation, external signals are ignored and BIST signals (B_ADDRx) are used.

B_EXTCLK is used to bring in the external clock signal to the BIST circuits for two purposes. The external clock is needed for the external clock option. It is also needed during the DFT mode to load in the address range (TLBADDRNG)

10.2 BOVBIAS

INPUTS	ESDA4, VUPB
OUTPUTS	B_OVDETECT
# OF CIRCUITS	1/chip
LOCATION	right corner

BOVBIAS is the overvoltage detector. Node OV will be charged up if there is an overvoltage condition on ESDA4 after VUPB goes low. This will set B_OVDETECT to high. A voltage higher than 5.2V on ESDA4 is sufficient for an overvoltage to be detected.

10.3 BIROSC (Fig 34)

INPUTS	PB_CLKBIST, BIST_EN, MD_SLFR, B_CLKMUX
OUTPUTS	B_IRCLK, BCLK_EN, B_CLK
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

The BIROSC is the primary oscillator that generates the clock signal for all BIST operation. It is a dual mode oscillator circuit. In BIST mode (when BIST_EN is high). It operates in the high frequency mode and provides B_CLK to the BIST circuit to synchronously control BIST operation. In normal mode (when BIST_EN) is low, it operates at the low frequency mode and produce a low frequency B_IRCLK used for self refresh mode. It is also a low frequency clock for the VBB circuits (Fig 66).

This circuit is also used to control internal and external clock for BIST operation.

If the device power up in BIST mode and the internal clock option is selected, B_CLKMUX will be set to low. This will set BCLK_EN to high to take over control of the CLK signals going into the device. The oscillator will generate 20ns clock needed for BIST operation. Externally CLK needs to be pulled low. If B_CLKMUX is high, external clock option is chosen, PB_CLKBIST will be used as the clock for BIST.

Nodes BNN and IRNN are used to compensate for process variation to ensure a more constant oscillator frequency.

10.4 BIRBIT (Fig 33)

INPUTS	CLK
OUTPUTS	Q
# OF CIRCUITS	6/chip
LOCATION	right corner of chip

This circuit is used to divide the CLK frequency by half. Q has half the frequency of CLK.

10.5 BSLFRCLK

INPUTS	B_IRCLK,VUPB
OUTPUTS	SLFR_TIME
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

SLFR_TIME is the clock used for self refresh and also in VBB pump. It oscillates with a period of about 8US.

10.9 BTIMEDRV

INPUTS	B_PCOUNTER(0)
OUTPUTS	B_PCOUNTERD(9:0), B_PCOUNTERDB (9:0)
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

Generate true and bar signals from the primary counter in the BIST circuit. The true and bar signals goes into the B_TGEN (Fig 47) circuit to generate the timing sets.

10.10 BRM (Fig 27)

INPUTS:B_ROM_ADDR (63:0)	
OUTPUTS: B_WORD(11:0)	
# OF CIRCUITS:	1/chip
LOCATION:	right corner of chip

The BRM has 64 ROM words and each word is 12 bit wide. The decoder in BIST circuit decides which ROM word is to be read.

10.10.1 Gross (Zrom) Algorithm

1. Write array with selected pattern.
2. Read array to confirm selected pattern was written.

Note:
a) Selected pattern is an all 0's pattern, an all 1's pattern or any combination of 0's and 1's.

10.10.2 Pause Algorithm

1. Write array with selected pattern.
2. Wait for specified time to elapse.
3. Read array to confirm selected pattern written is retained.

Note:
a) Selected pattern is an all 0's pattern, an all 1's pattern and/or any combination of 0's and 1's.

10.10.3 Xmarch Algorithm

1. Write array with background pattern.
2. For each row read the background pattern and write the inverted pattern into each column.
3. For each row read the inverted pattern and write the original pattern into each column.
4. Read original background pattern to confirm no defect has occurred.

Notes:
a) Background pattern is an all 0's pattern, an all 1's pattern or combination of 0's and 1's.

10.10.4 Ymarch Algorithm

1. Write array with background pattern.
2. For each column read the background pattern and write the inverted pattern into each row.
3. For each column read the inverted pattern and write the original pattern into each row.

4. Read original background pattern to confirm no defect has occurred.

Notes:
a) Background pattern is an all 0's pattern, an all 1's pattern or combination of 0's and 1's.

10.10.5 Sdist Algorithm

1. Write array with background pattern (optional)
2. Write target row with disturb pattern repeatedly until specified time has elapsed.
3. Refresh
4. Read neighboring rows to target row to confirm no disturb type error defect has occurred.
5. Return background pattern to target row.
6. Repeat by targeting all rows in the chip.

Notes:

a) Disturb pattern is an all 1's pattern, an all 0's pattern or any combination of 1's and 0's.
b) Background pattern is an all 0's pattern, an all 1's pattern or combination of 0's and 1's.
c) Steps 4 and 5 can be interchanged.

10.10.6 Ldist Algorithm

Algorithm steps are the same as for Sdist Algorithm. The Ldist algorithm utilizes a different time set to lengthen the time between a RASB and CASB pulses.

10.10.7 Pdist Algorithm

Algorithm steps are the same as for Sdist Algorithm. The Pdist algorithm writes the disturb pattern in a page mode. In the page mode the row remains activated until all columns are written into the row.

10.10.8 Burnin Algorithm

1. Continually write array with selected pattern.
Note:
a) Selected pattern is an all 0's pattern, an all 1's pattern or any combination of 0's and 1's.

10.10.9 Write 1 ROM Algorithm

1. Write one row in the array with selected pattern.
Note:
a) Selected pattern is an all 0's pattern, an all 1's pattern or any combination of 0's and 1's.

10.10.10 Read 1 Column Algorithm

1. Read one column in the array.
Note:
a) Selected pattern is an all 0's pattern, an all 1's pattern or any combination of 0's and 1's.
10.10.11 Finish Algorithm Set the DONE flag to true and deactivate the program counter.

10.11 to 10.18 BRM0_7 to BRM56_63

INPUTS	B_ROM_ADDRx where x ranges from 0 to 63
OUTPUTS	B_WORD0 to B_WORD11
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

The BIST ROM circuits is divided into 8 individual circuits with each circuit having 8 words.

10.19 BROMDRV (Fig 53)

INPUTS	B_WORD,B_PRECHARGE,B_ROM_WORD
OUTPUTS	B_WORD0..B_WORD11
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

BROMDRV is a driver circuit for the 12 bits of a ROM word. The 12 bits of data line is precharged once every 2 clock cycle during BIST operation when the device is not in the array access mode (Fig 68).

10.20 BIST (Fig 51,52)

INPUTS:
BIST_EN, B_EEPRMOPT, TLBADDRNG, PB_CLKBIST, B_SR_LOAD,

B_YSTOPE, B_XSTOPE, B_YSTARTE, B_XSTARTE, VUPB, TLBMON3
TLBMON2, TLBMON1, TLBROMR, B_ROM_WORD
B_RASA1, B_RASB1, B_RASC1, B_RASA2, B_RASB2, B_RASC2,

B_CASA1, B_CASB1, B_CASC1, B_CASA2, B_CASB2, B_CASC2,
B_OEA1, B_OEB1, B_OEC1, B_OEA2, B_OEB2, B_OEC2,

B_DMUXA1, B_DMUXB1, B_DMUXC1, B_DMUXA1, B_DMUXB1, B_DMUXC1,

B_YMUXA1, B_YMUXB1, B_YMUXC1, B_YMUXA1, B_YMUXB1, B_YMUXC1
B_FIXCOUNTA, B_FIXCOUNTB, B_FIXCOUNTC
B_PERIODA1, B_PERIODB1, B_PERIODC1,

OUTPUTS: B_DONE, B_OUTPUTEN, B_CLKMUX, B_ROM_PRECHARGE,
B_DQL(23:18), B_RASB, B_CASB, B_WDATA_TIMB, B_WB,
B_PCOUNTER, B_ADDR, B_PF_RESULT, B_ROM_ADDR, B_DQ

OF CIRCUITS: 1/chip

LOCATION: right corner of chip

All circuits in this block is synthesized using Autologic2 after coding in VHDL code. It has 8 major blocks :

1. B_CLK_GEN (Fig 22): Generates auxiliary clocks of different phases. Also generates the precharge signal for the ROM
2. B_ADDRCTL (Fig 8-12): Generates and control the addresses of the cells to be accessed
3. B_CROM (FIG 40) : Stores the test conditions in the ROM, holds the program counter and decoder
4. B_PG_CONTROL (Fig 24): Controls the flow of test program.
5. B_PASSFAIL : Generates the data pattem to be written to array, perform pass fail comparison.

6. B_DQMUX : Control the multiplexing for BIST signals to appear at the DQs.
7. B_ADDRNG (Fig 16-21): Control the start and stop address of BIST operation.
8. B_TGEN (Fig 47-50) : Control and generates the timing of all the Control, Data, and address signals.

10.21 B_CLK_GEN

INPUTS	B_CLK, VUPB,B_SR_LOAD,B_DONE, B_ROM_PRE_EN,
OUTPUTS	B_CLK_B,B_RESET,B_CLK_C,B_CLK_A,B_ROM_PRECHARGE, B_CLK_AD
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

This circuit takes the BIST primary clock B_CLK and makes 4 other clocks out of it in 2 clock cycles. Rising edge of B_CLK_A updates the newest instruction pointed to by the Program counter . Decoding of the newest instruction starts at this edge. Rising edge of B_CLK_AD is the clock that executes the newly decoded instruction. At the rising edge of B_CLK_B, a decision is made on the action to take with the program counter. B_CLK_C acts as a reset signal.

It also generates the precharge signal for the ROM (Fig 69).

10.22 B_CROM (FIG 40)

INPUTS: B_CLK_A,B_CLK_C,B_ROM_WORD,TLBROMR,B_RESET,
B_INS_COMPLETED,B_PC_LOAD,B_CLK_B,B_PC_LO_ADDR,
B_RPTINV_STATE

OUTPUTS:
B_RPT_N_INV,B_YEN,B_DONE,B_DECR,B_ALTERNATE,B_INC,
B_XEN,B_TIME_SET1,B_XEN,B_TIME_SET0,B_CKBD,B_CHK_KEY,
B_REN,B_READ,B_PAUSE,B_JUMP,B_CHK_TIMEOUT,B_WRITE,
B_JMP_ADDR,B_ROM_ADDR,B_RDATA,B_WDATA

OF CIRCUITS: 1/chip

LOCATION: right corner of chip

B_CROM controls the operations of the ROM. It is made up of a decoder, a program counter and B_ROMLOGIC (Fig 30) which decodes what the current instruction. The program counter points to word 0 during power up. Subsequently in BIST operation. PC will point to the relevant word in the ROM to execute the desired instruction. The decoder decodes the 6 bit program counter to point to one of the 64 words in the ROM (Fig 70).

10.23 B_DECODER (Fig 26)

INPUTS	B_PC_ADDR
OUTPUTS	B_ROM_ADDR
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

This is a simple 6 to 64 decoder. 6 bit input address from the program to be decoded into 64 words in the ROM

10.24 B_PC (Fig 25)

INPUTS	B_PC_LOAD,B_PC_LD_ADDR,B_CLK_B, B_RESET, B_INS_COMPLETED
OUTPUTS	B_PC_ADDR,
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

The program counter points to an instruction in the ROM to be executed. B_CLK_B is used to change the program counter to its new value.

At the completion of an instruction, program counter can change in 2 ways. It can either be incremented by 1 (normal program flow), or it can jump to any one of the 64 words of the ROM(conditional or unconditional jump).

At the rising edge of B_CLK_B, the circuit will look at two signals that comes in: B_PC_LOAD and B_INS_COMPLETED.

If B_PC_LOAD is a high, it will do a jump by loading the PC with the address that appears on B_PC_LD_ADDR. If B_INS_COMPLETED is a high, it will increment the PC by 1 and proceed to the next instruction. If neither B_PC_LD_ADDR nor B_INS_COMPLETED is high then no action will be taken and the PC will remain the same. B_PC_LOAD and B_PC_LD ADDR cannot be high at the same time (Fig 71).

If one wants to read the content of the ROM and output it to the DQ pins, ROM read DFT mode can be used. The device needs to be powered up with overvoltage, then perform a DFT entry to TLBROMR while in BIST standby mode, then pull CS high to go into BIST mode.

The external clock is used to move from one ROM word to the next ROM word ,The PC is incremented every 2 clock cycles. i.e. on every rising edge of B_CLK_B. This is done by forcing B_INS_COMPLETED to be high all the time during the DFT TLROMR mode.

10.25 B_ROMLOGIC (Fig 30,31)

INPUTS: B_ROM_WORD, TLBROMR, B_CLK_C, B_CLK_A, B_RESET

OUTPUTS: B_JMP_ADDR, B_DONE, B_CHK_TIMEOUT, B_RPT_N_INV, B_TIMESET1, B_TIMESET0, B_DECR, B_XEN, B_JUMP, B_CHK_KEY B_ALTERNATE, B_REN, B_WRITE, B_PAUSE, B_READ, B_RDATA, B_WDATA, B_CKBD, B_INC, B_YEN, B_CLK_ARESET

OF CIRCUITS: 1/chip

LOCATION: right corner of chip

B_ROMLOGIC decodes the 12 bits of data contained in an instruction word. There are two major types of instructions. First is the program control instruction. This type of instruction deals with the flow of the program. They are introduced to control BIST operation. Second is the array access instruction and they control how the array is to be tested .They are basically the same type of instruction usually found on a tester. They are translated into BIST format in this case.

An instruction is divided into two parts. The first six bits of an instruction defines the actions to be taken and the last six bits is the data associated with the instruction.

For an read whole array instruction, Read(bitll),X(bit9) and Y(bit8) are set to 1. The last six bits of the instruction provides information on how the read whole array is to be achieved. i.e. timing sets to be used, data pattern to be used, etc.

For a program control instruction, The 4 most significant bits are '0', Bit 7 and Bit 6 determines the program control instruction. The last six bits provides the address to be jumped to if indeed the decision is made to jump.

There is an unconditional jump instruction. '110011' for the most significant 6 bits, and the address to jump to for

the 6 least significant bits.

The last instruction in the ROM is an idle instruction to signal the end of BIST operation. The last six bits of this instruction holds the revision number of the current 256M (Fig 72).

The instruction above will perform a read, from the whole array (both X and Y enabled), using timing set A, expected data '0', true internal data pattern, and same data among the DQ.

There are many possible combination of instructions that can be programmed with the current circuits. If a new algorithm is need for BIST. It can be included by simply reprogramming the ROM. Combinations of options are available to make up an instruction (Fig 73)

10.26 B_TGEN

```
        INPUTS: B_READ,B_RESET,B_PAUSE,B_YCARRY_1,B_CLK,B_WRITE,
                B_TIME_SET0,B_TIME_SET1,B_INS_COMPLETED,B_REN,
                B_OEA1,B_OEA2,B,B_OEB1,B_OEB2,B,B_OEC1,B_OEC2,
                B_RASA1,B_RASA2,B_RASB1,B_RASB2,B_RASC1,B_RASC2,
                B_CASA1,B_CASA2,B_CASB1,B_CASB2,B_CASC1,B_CASC2,

B_DMUXA1,B_DMUXB1,B_DMUXC1,B_DMUXA1,B_DMUXB1,B_DMUXC1,

B_YMUXA1,B_YMUXB1,B_YMUXC1,B_YMUXA1,B_YMUXB1,B_YMUXC1
                B_FIXCOUNTA,B_FIXCOUNTB,B_FIXCOUNTC
                B_PERIODA1,B_PERIODB1,B_PERIODC1,
       OUTPUTS: B_YSELB,B_RASB,B_CASB,B_WB,B_OEB,B_PERIODB,

B_ROM_PRE_EN,B_WDATA_TIMB,B_TIMECNT_REF,B_PCOUNTER
 # OF CIRCUITS: 1/chip
      LOCATION: right corner of chip
```

The timing sets used in BIST tests are generated through the use of the primary clock generated in BIROSC (Fig 34) block. The clock is a 50MHz clock and the resolution of the timing sets is 20ns. A counter counts the number of rising edges of the primary clock. The control timing is set by specifying a specific count of the counter. In the example below, Activate is specified at count 2 and Deactivate is specified at count 12 (Fig 74).

Altogether 3 types of timing sets, namely short, long and page timing set were used in the 10 algorithms implemented in BIST (Fig 75) .

10.27 B_COUNTER10B (FIG 39)

INPUTS	B_CNTR10_CLK, B_CNTR10_CLR
OUTPUTS	B_CNTR10
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

Rising edge of B_CNTR10_CLK will increment the counter by 1. The counter is always enabled.

10.28 B_COUNTER10A (FIG 38)

INPUTS	B_CNTR10_CLK,B_CNTR10_EN,B_CNTR10_CLR
OUTPUTS	B_CNTR10
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

If B_CNTR10_EN is high, rising edge of B_CNTR10_CLK will increment the counter by 1

10.29 B_ADDRCTL

INPUTS	B_PAUSE,B_PERIODB,B_RESET,B_CLK_AD, B_YEN,B_XEN, B_REN,B_INC,B_YSELB, BIST_EN,B_DECR,B_SUBARRAY, B_XADDR_STP,B_YADDR_STP,B_XADDR_STR, B_YADDR_STR
OUTPUTS	B_YCLR,B_XCLR,B_RCARRY,B_YCARRY,B_XCARRY, B_YNOCARRYJ, B_YCARRY_1,B_XNOCARRY,B_LSB_YADDR, B_LSB_XADDR,B_ADDR
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

B_ADDRCTL has two blocks, B_ADDR_CTR which contains all the address counters and B_RESET_EN which provide controls to detect overflows of address counters .

10.30 B_ADDR_CTR

INPUTS:	B_SUBARRAY, B_INC, B_DECR, B_ADDRCLK, B_XCLR, B_YCLR, B_RCLR, B_XEN, B_YEN, B_REN, B_YSELB, BIST_EN, B_XADDR_STP, B_YADDR_STP, B_XADDR_STR, B_YADDR_STR
OUTPUTS:	B_RCARRY, B_YCARRY, B_XCARRY, B_YNOCARRYJ, B_YCARRY_1, B_XNOCARRY, B_LSB_YADDR, B_LSB_XADDR, B_ADDR
# OF CIRCUITS:	1/chip
LOCATION:	right corner of chip

B_ADDR_CTR has 3 counters, the row counter, the column counter and the refresh counter. A multiplexer decides which counter value is used as the B_ADDR signal during BIST operation. In normal access, only the X counter and Y counter are used. B_YSELB is the signal used to choose between the column and row address (Fig 76). During refresh, only the Refresh counter is used.

The addressing MUX sequence and controlled by a combination of B_ADDRCLK, B_INC and B_YSELB.. B_PERIODB pulse generates B_ADDRCLK which increment the address counter by 1.

Low on B_YSELB indicates selection of column address. High on B_YSELB indicates selection of row address. B_INC will increment the enabled address counter by 1 (Fig 77).

10.31 B_ROW_CTR (Fig 46,55)

INPUTS: B_INC, B_DECR, B_ADDRCLK, B_XCLR, B_XEN, B_XSTART, B_XSTOP
OUTPUTS: B_CARY, B_XCARRY, B_XNOCARRYJ, B_XADDR
OF CIRCUITS: 1/chip
LOCATION: right corner of chip

The X counter is a 14 bit counter and it holds the current row address. It can count forward and backward. depending the state of B_DECR. A low on B_DECR means counting forward , i.e. increment the counter value by 1 with every rising edge of the B_ADDRCLK. The counter is enabled by B_XEN.

If the subarray option is chosen. The counter will be start with the start address loaded in earlier. An overflow will be issued once the stop address is reached.

10.32 B_COL_CTR (Fig 37)

INPUTS: B_INC, B_DECR, B_ADDRCLK, B_YCLR, B_YEN, B_YSTART, B_YSTOP
OUTPUTS: B_CARY, B_YCARRY, B_YNOCARRYJ, B_YADDR
OF CIRCUITS: 1/chip
LOCATION: right corner of chip

The Y counter is a 9 bit counter and it holds the current column address. It can count forward and backward. depending the state of B_DECR. A low on B_DECR means counting forward , i.e. increment the counter value by 1 with every rising edge of the B_ADDRCLK. The counter is enabled by B_YEN.

If the subarray option is chosen. The counter will be start with the start address loaded in earlier. An overflow will be issued once the stop address is reached.

10.33 B_REF_CTR (Fig 45)

INPUTS	B_ADDRCLK,B_RCLR,B_REN,B_XSTART,B_XSTOP,
OUTPUTS	B_RCARRY,B_RADDR
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

The refresh counter is a 14 bit counter. It only counts forward. The counter value is incremented by 1 with every rising edge of the B_ADDRCLK if B_REN is high. The refresh counter is used in the three disturb tests only. During the refresh instruction, a pseudo read is done for all the rows that are enabled. This refreshes the array but no pass/fail comparison is done.

10.34 B_ADDR_RESET_EN (Fig 35,36,54)

INPUTS: B_INC, B_PERIODB, B_XEN, B_YEN, B_REN, B_RCLR, B_REN, B_X2YLINK
B_RESET, B_CLK_C, B_CLK_AD, B_XCARRY, B_YCARRY, B_RCARRY, B_PAUSE
OUTPUTS: B_ADDRCLK, B_YENABLE, B_XRESET, B_YRESET, B_RRESET
OF CIRCUITS: 1/chip
LOCATION: right corner of chip

This circuit generates the clock (B_ADDRCLK) used to increment or decrement the address counters. B_ADDRCLK is generated in 2 ways namely during an INC instruction or during the end of an array-access cycle. The reset signals (XCLR, YCLR and RCLR) are generated at the rising edge of B_CLK_C. These reset the counters to its original state when overflows occur.

B_X2YLINK is used to join the X and Y register together for whole array tests. The CARRY signal is used to signal the completion of an array access instruction. It is sent to the program counter to tell it to move on to the next instruction.

10.35 B_PG_CONTROL

INPUTS: VUPB, B_RESET, BIST_EN, B_DONE, B_TIMEOUT_CHK, B_INC,
B_XEN, B_XCARRY, B_XNOCARRYJ, B_XCLR, B_YEN, B_YCARRY, B_YNOCARRYJ, B_YCLR, B_REN, B_RCARRY, B_CHK_KEY,
B_RPT_N_INV, B_JUMP, B_TIMECNT_REF, B_ADDRU, B_ADDRL, B_CLK_C, B_CLK_A, B_SR_LOAD
OUTPUTS:
B_INS_COMPLETED, B_PC_LOAD, B_RPTINV_STATE, B_SUBARRAY B_CLKMUX, B_OUTPUTEN
OF CIRCUITS: 1/chip
LOCATION: right corner of chip

This block contains the circuit used for controlling the Program counter. It also contain a shift register which holds information on which test to perform.

Upon completion of an instruction, this block will make a decision on whether to increment the program counter or to load the program counter with a new address.

10.36 B_SHIFT (Fig 7)

INPUTS	VUPB,BIST_EN,B_DONE,B_SR_CLK,B_SR_LOAD,B_ADDRU, B_ADDRL
OUTPUTS	B_LSB_SR,B_SUBARRAY,B_CLKMUX,B_OUTPUTEN

(continued)

# OF CIRCUITS	1/chip
LOCATION	right corner of chip

B_SHIFT register holds the data latched in during power up. Every time the test_en instruction is executed once, the shift register is shifted right by 1 position. This instruction looks at the last bit to see if the test is enabled (Fig 78).

10.37 B_ECOMPARE10 (Fig 43)

INPUTS	B_ECMPR10_EN,B_AIN9_0,B_BIN9_0
OUTPUTS	B_ECMPR10_RSLT,B_ECMPR10_RSLTB
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

This is a simple comparator. If the comparison enable signal is high, then B_ECMPR10_RSLT will be high and the bar signal will be low if the 2 10 bit inputs are equal. Alternatively, if the 2 inputs are not equal,B_ECMPR10_RSLT will be low. If the comparison is not enabled, then both outputs will be low regardless of the inputs.

10.38 B_PASSFAIL (Fig 33,44)

INPUTS	B_RESET,B_CLK,B_RDATA,B_WDATA,B_OEB,B_ALTERNATE, B_CKBD, B_WRITE, B_LSB_XADDR,B_LSB_YADDR, B_DQ
OUTPUTS	B_WEVNVAL,B_WODDVAL,B_ PF_RESULT,B_DODD,B_DEVEN
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

This circuit generates internal data pattern to be written to the array. It also generates the expected data for pass fail comparison. The timing for comparison is controlled by B_OEB timing.

10.39 B_DQMUX (Fig 41,42)

INPUTS	TLBROMR,TLBMON1,TLBMON2, TLBMON3,B_WRITE, B_WDATA_TIMB,B_ROM_WORD, B_WEVNVAL,B_WODDVAL, B_RASB,B_CASB,B_WB,B_CLK,B_DODD,B_DEVEN,B_OEB, B_ADDR
OUTPUTS	B_DQ,B_DQL
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

This circuits is a huge multiplexer to multiplex the different signals that goes out to the outside world. In the normal mode, only the pass fail signals goes to the outside. In the other three monitor modes, different control signals are brought to the outside (Fig 79).

10.40 B_ADDRNG

```
         INPUTS:
B_EEPRMOPT,PB_CLKBIST,TLBADDRNG,B_ADDR,B_XSTARTE,
                    B_XSTOPE,B_YSTARTE,B_XSTOPE,
        OUTPUTS: B_XADDR_STR,B_XADDR_STP,B_YADDR_STR,B_YADDR_STP
 # OF CIRCUITS: 1/chip
       LOCATION: right corner of chip
```

This circuit is used to load start and stop address for sub array BIST testing. A DFT mode is used to get into this situation. When this DFT mode is entered, the next 4 rising edges of the clock will load in the X and Y start and stop address (Fig 80).

10.41 BONEXX

INPUTS	IN
OUTPUTS	OUT
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

This is the '1' bit of the ROM and BONExx represents a high. IN is connected to the 'wordlines' (B_ROM_ADDR(x)). OUT is always precharged to high and does not get pulled to low whether IN is high or not (Fig 81).

10.42 BZEROXX

INPUTS	IN
OUTPUTS	OUT
# OF CIRCUITS	1/chip
LOCATION	right corner of chip

This is 0 bit of the ROM and BZEROxx represents a low. IN is connected to the 'wordlines' (B_ROM_ADDR(X)). OUT (B_WORD(Y)) is precharged to high but if this cell is selected and IN goes high, OUT will be pulled to low (Fig 82)

The foregoing describes the arrangement and operation of an exemplary integrated circuit memory device having built-in self-test circuitry. The described arrangement and method of and other arrangements made obvious in view there of are considered to be within the scope of the appended claims.

Claims

1. An integrated circuit having a built-in self-test (BIST) arrangement, the built-in self-test arrangement comprising:

   a read only memory for storing test algorithm instructions;
   a logic circuit, interconnected with the read only memory for receiving an instruction read from the read only memory and for producing a group of output signals dependent upon the instruction at an output terminal thereof;
   a BIST register connected with the output terminal for receiving and storing the group of output signals from the logic circuit.

2. The integrated circuit as claimed in Claim 1 further comprising:
   a timing generator interconnected with the BIST register, for receiving signals stored in the BIST register and

for producing row address strobes, column address strobes and a write enable signal.

3. The integrated circuit as claimed in Claim 2 further comprising:

a multiplexer having plural groups of input terminals, a group of input control terminals, and a group of output terminals;
a first bus interconnecting a first group of input terminals of the multiplexer to the BIST register for conveying signals stored in the BIST register to the multiplexer; and
a second bus interconnecting the group of input control terminals of the multiplexer for transmitting the row address strobes, the column address strobes, and the write enable signal to the multiplexer.

4. The integrated circuit as claimed in Claim 3 further comprising:

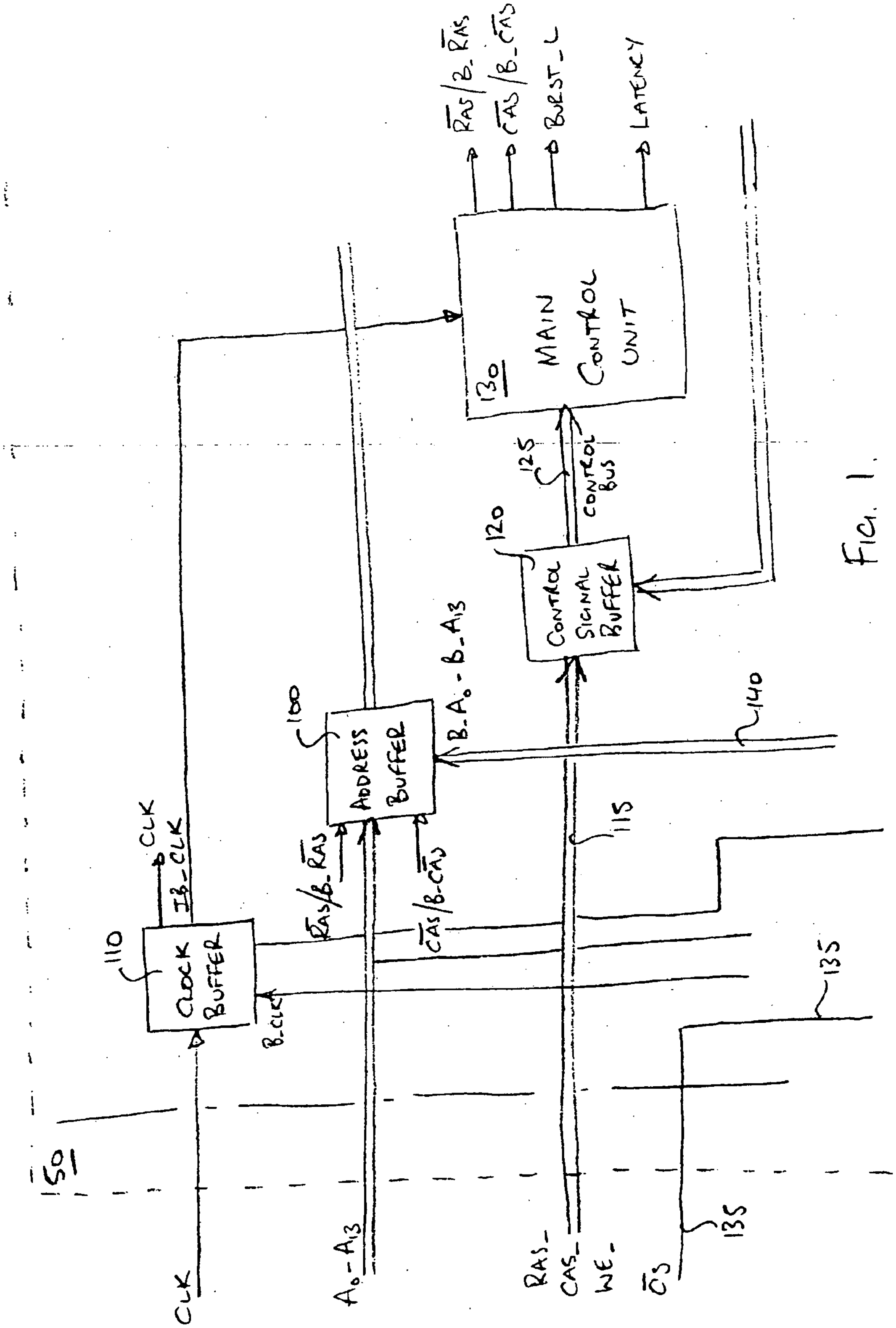
an oscillator for producing a BIST clock signal;
a BIST program counter arranged for counting in response to the BIST clock signal;
a third bus interconnecting a group of input terminals of the BIST program counter to the BIST register for conveying signals stored in the BIST register to the BIST program counter;
a BIST address decoder responsive to a count in the BIST program counter for decoding the count and producing a signal for selecting a row address for the read only memory; and
a fourth bus interconnecting row address inputs of the read only memory to the BIST address decoder for conveying the decoded row address from the BIST address decoder to the row address inputs of the read only memory.

5. The integrated circuit as claimed in Claim 4, wherein the BIST program counter comprises;

a reset input for receiving a reset signal for setting the BIST program counter to a predetermined initial state;
an increment input for receiving an increment of the counter signal for incrementing a count state stored in the BIST program counter; and
a jump input for receiving a jump signal for causing the state of the BIST program counter to change to a state represented by signals conveyed from the BIST register.

6. An integrated circuit memory array device having a built-in self-test (BIST) arrangement, the memory device comprising:

an address buffer for receiving and storing a sequence of addresses;
a built-in self-test detector responsive to input signals for controlling operation of the device alternatively in either a normal mode or a self-test mode;
address range registers arranged for receiving and storing start and end addresses for a self-test operation of the memory array while the device is operating in the self-test mode; and
a self-test address counter interconnected with the address range registers and the address buffer for applying a sequence of addresses commencing with the start address and stopping at the end address to the address buffer.



CLK
A₀-A₁₃
RAS_
CAS_
WE_
CS
150
135
110
CLOCK BUFFER
CLK
IB_CLK
B_CLK
100
ADDRESS BUFFER
B_A₀ - B_A₁₃
115
140
120
CONTROL SIGNAL BUFFER
125
CONTROL BUS
130
MAIN CONTROL UNIT
BURST_L
LATENCY

FIG. 1.

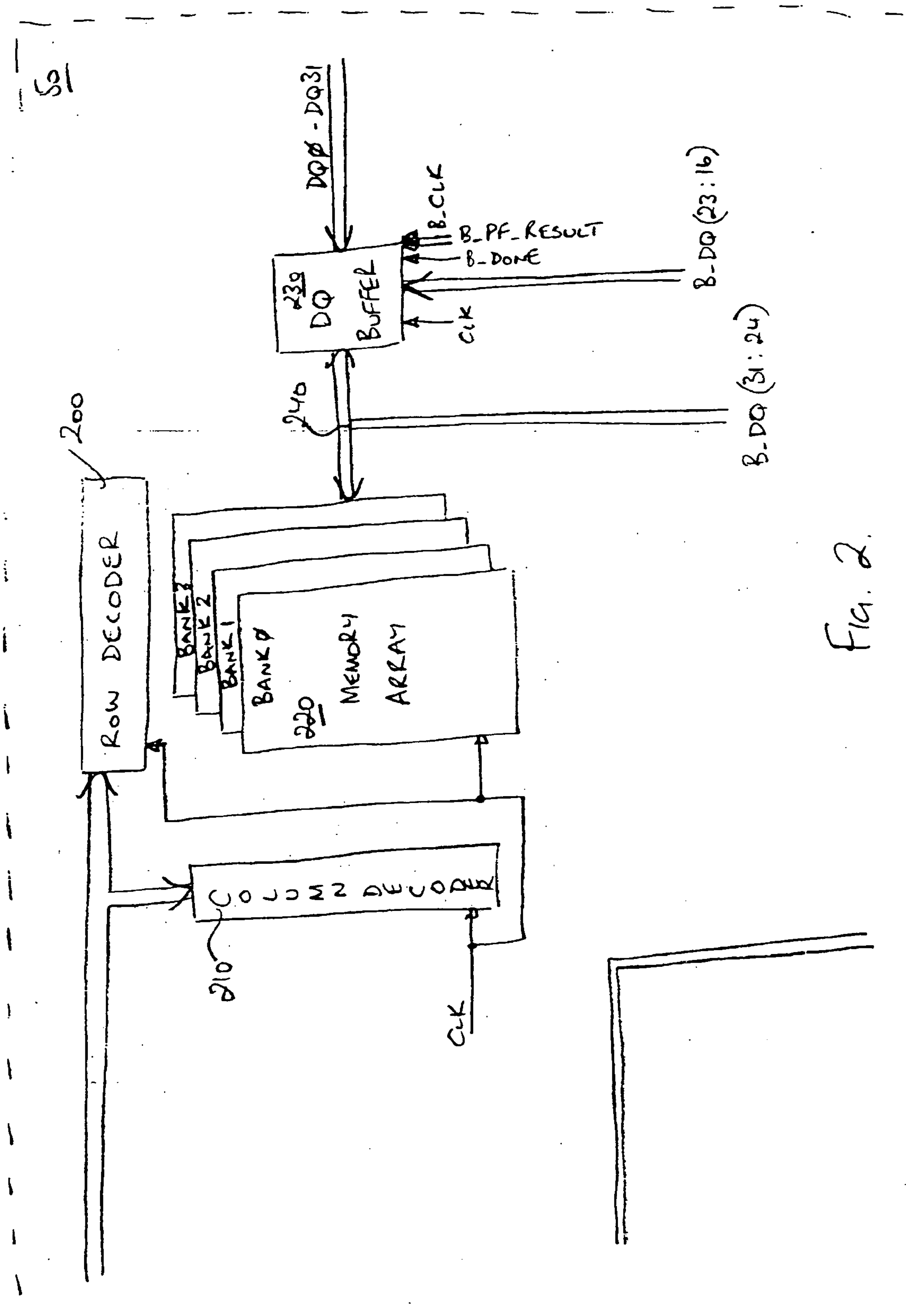

231
DQØ - DQ31
B_CLK
B_PF_RESULT
B_DONE
B_DQ (23:16)
230
DQ
BUFFER
CLK
B_DQ (31:24)
240
200
ROW DECODER
BANK 3
BANK 2
BANK 1
BANK Ø
220
MEMORY
ARRAY
COLUMN DECODER
210
CLK

FIG. 2.

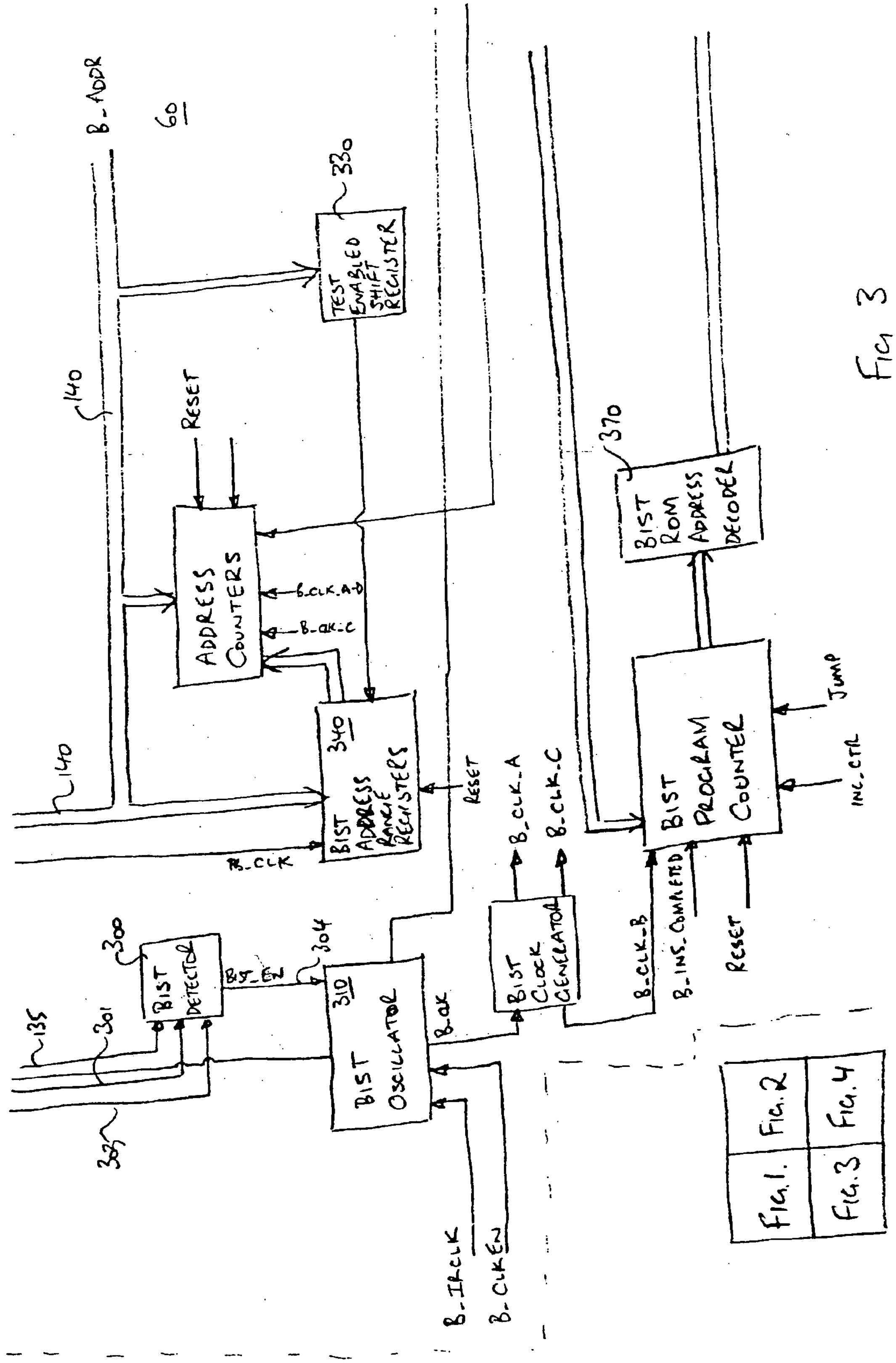

B_ADDR
60
330
TEST ENABLED SHIFT REGISTER
140
RESET
ADDRESS COUNTERS
B_CLK_A-D
B_CLK_C
370
BIST ROM ADDRESS DECODER
140
340
BIST ADDRESS RANGE REGISTERS
RESET
B_CLK
BIST PROGRAM COUNTER
JUMP
INC_CTR
B_CLK_A
B_CLK_C
300
BIST DETECTOR
BIST_EN
304
310
BIST OSCILLATOR
B_CLK
BIST CLOCK GENERATOR
B_CLK_B
B_INS_COMPLETED
RESET
135
301
302
B_IRCLK
B_CLKEN

FIG. 3

FIG. 1.
FIG. 2
FIG. 3
FIG. 4

FIG. 5

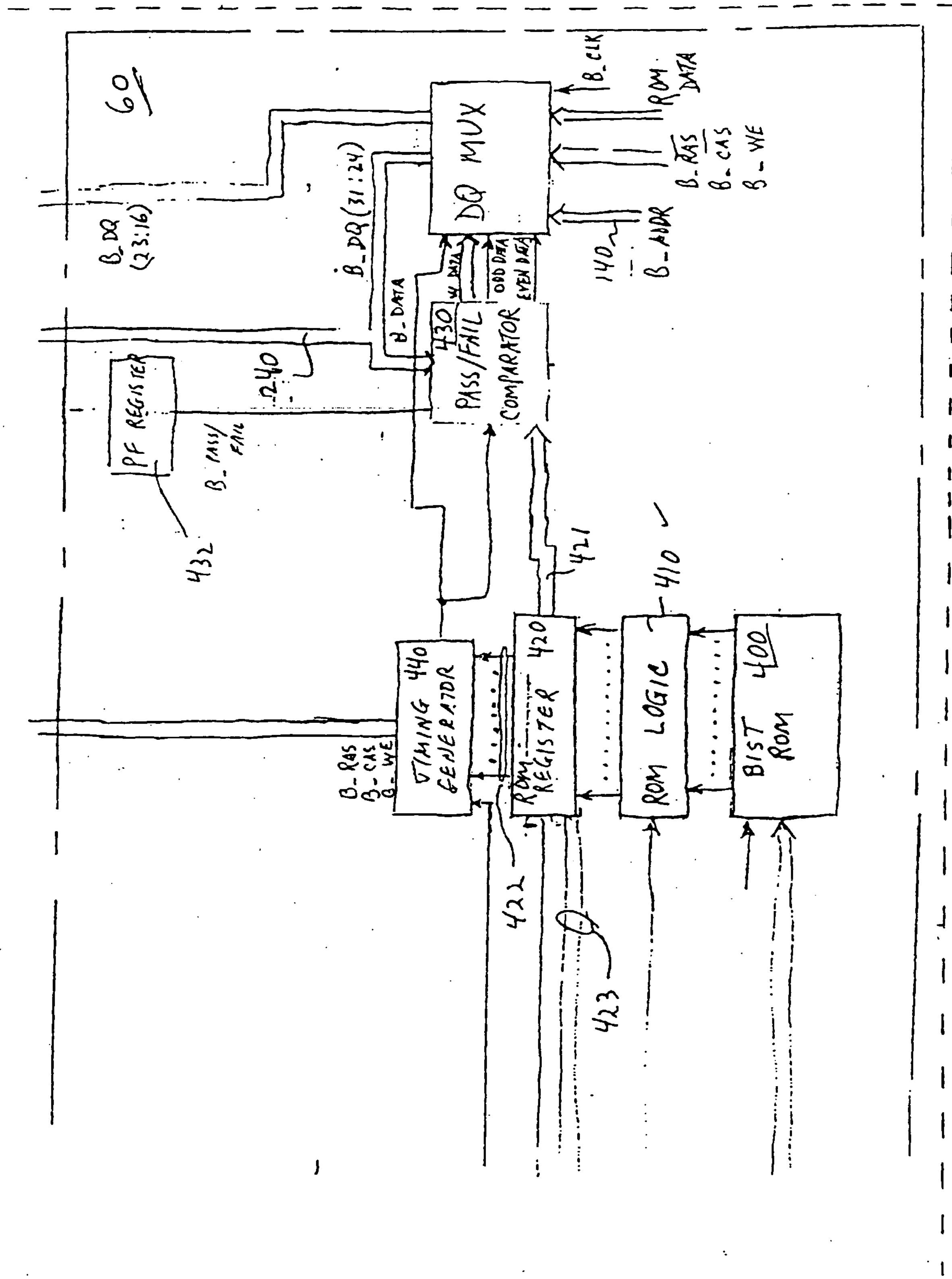

60
DQ MUX
B_CLK
ROM DATA
B_RAS
B_CAS
B_WE
B_ADDR
140
B_DQ(31:24)
B_DQ(23:16)
B_DATA
W DATA
ODD DATA
EVEN DATA
430
PASS/FAIL COMPARATOR
240
PF REGISTER
B_PASS/FAIL
432
421
410
420
TIMING 440 GENERATOR
B_RAS
B_CAS
B_WE
ROM REGISTER
ROM LOGIC
BIST ROM 400
422
423

FIG. 4

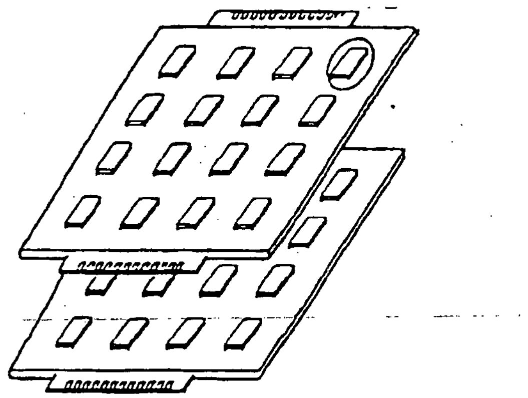


FIG. 6

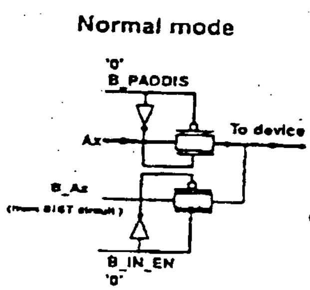


FIG. 13A

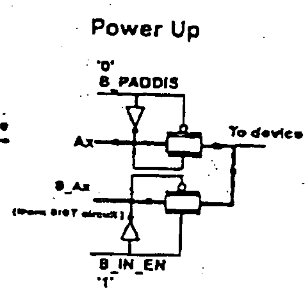


FIG. 13B

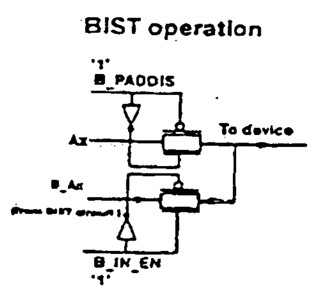


FIG. 13C

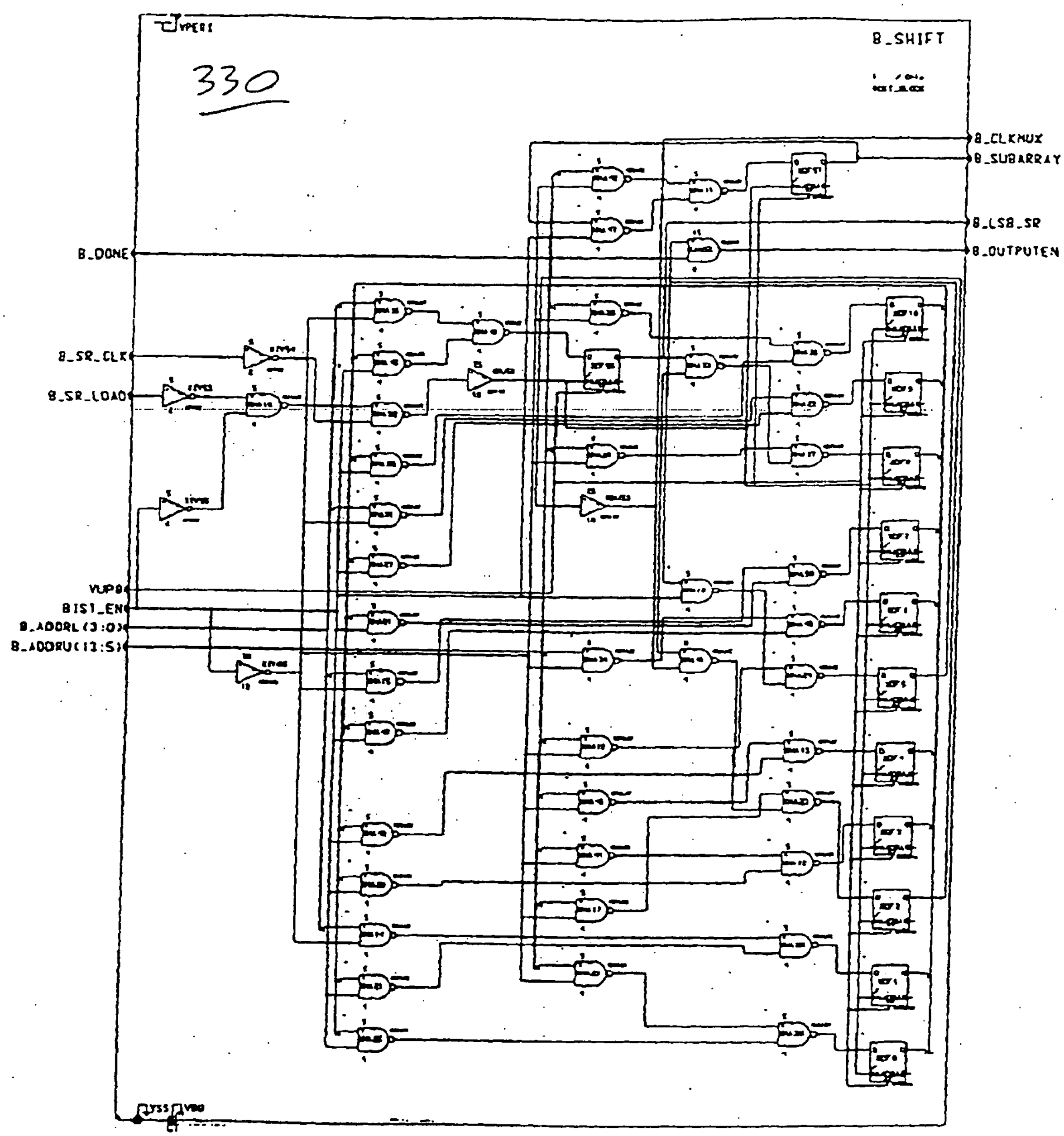

330
B_SHIFT
B_CLKMUX
B_SUBARRAY
B_LSB_SR
B_OUTPUTEN
B_DONE
B_SR_CLK
B_SR_LOAD
VUPB
BIST_EN
B_ADDRL(3:0)
B_ADDRU(13:5)

FIG. 7

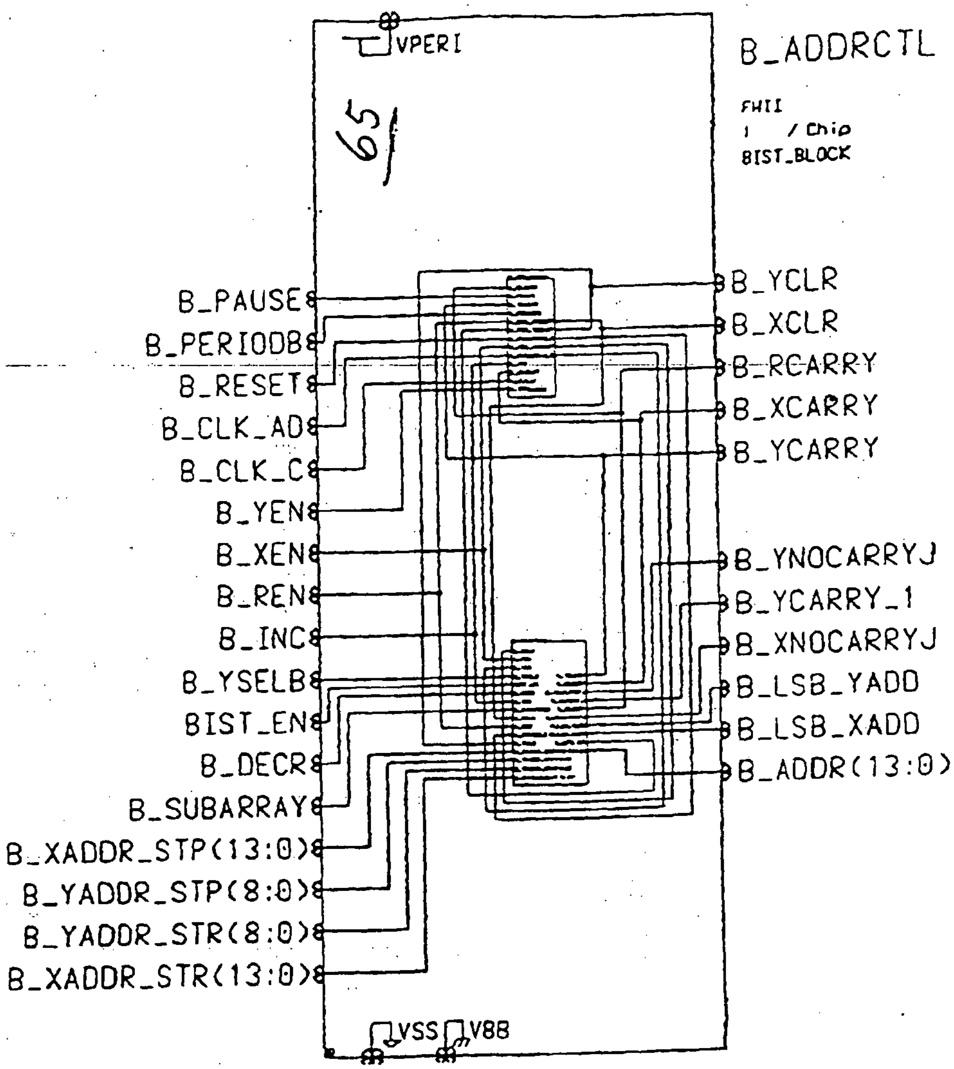

VPERI
B_ADDRCTL
FHII
1 / Chip
BIST_BLOCK
65
B_PAUSE
B_PERIODB
B_RESET
B_CLK_AD
B_CLK_C
B_YEN
B_XEN
B_REN
B_INC
B_YSELB
BIST_EN
B_DECR
B_SUBARRAY
B_XADDR_STP(13:0)
B_YADDR_STP(8:0)
B_YADDR_STR(8:0)
B_XADDR_STR(13:0)
B_YCLR
B_XCLR
B_RCARRY
B_XCARRY
B_YCARRY
B_YNOCARRYJ
B_YCARRY_1
B_XNOCARRYJ
B_LSB_YADD
B_LSB_XADD
B_ADDR(13:0)
VSS
VBB

FIG. 8

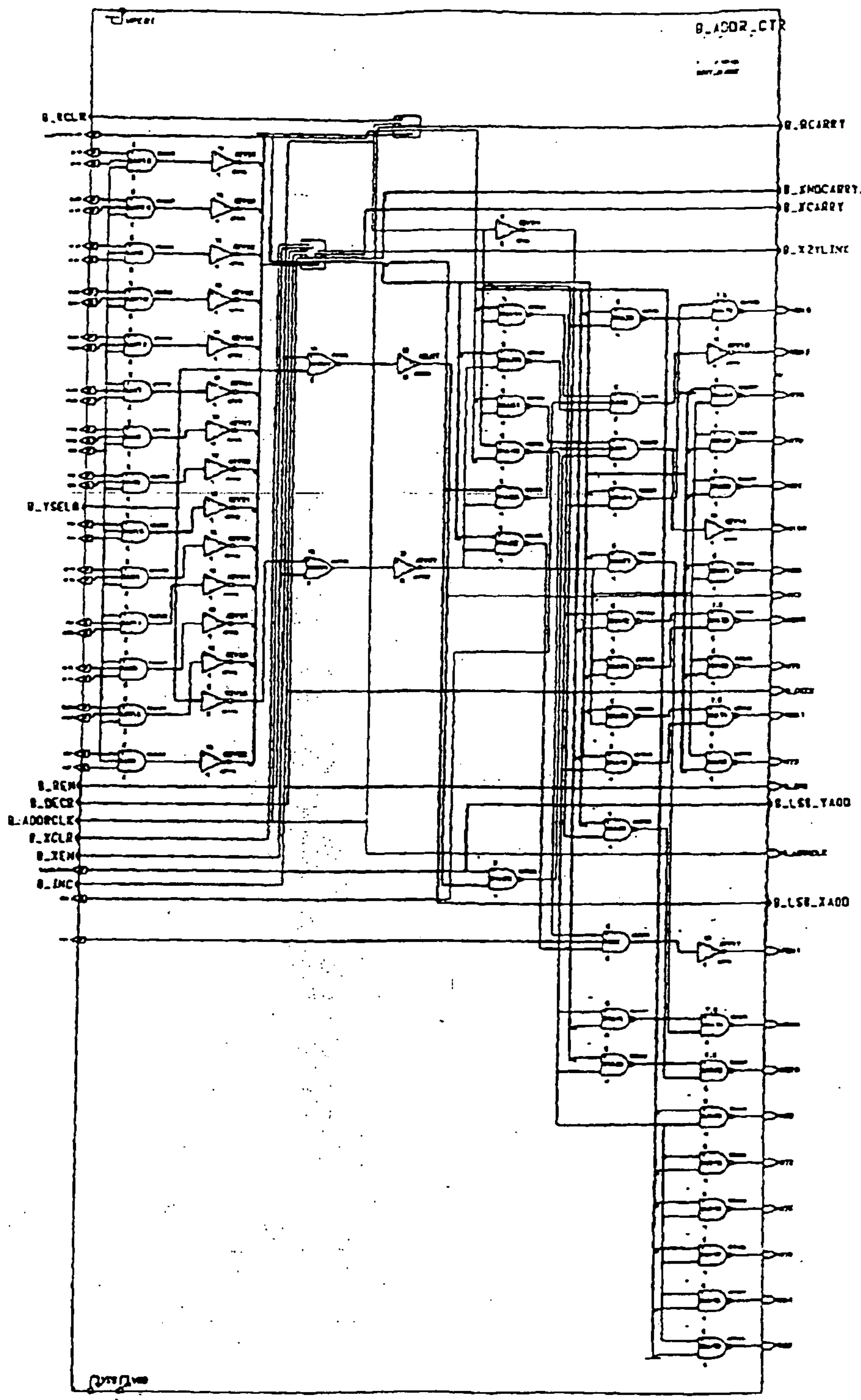

B_ADDR_CTR
B_RCARRY
B_XCARRY
B_X2YLINC
B_YSELB
B_LSB_YADD
B_LSB_XADD

FIG. 9

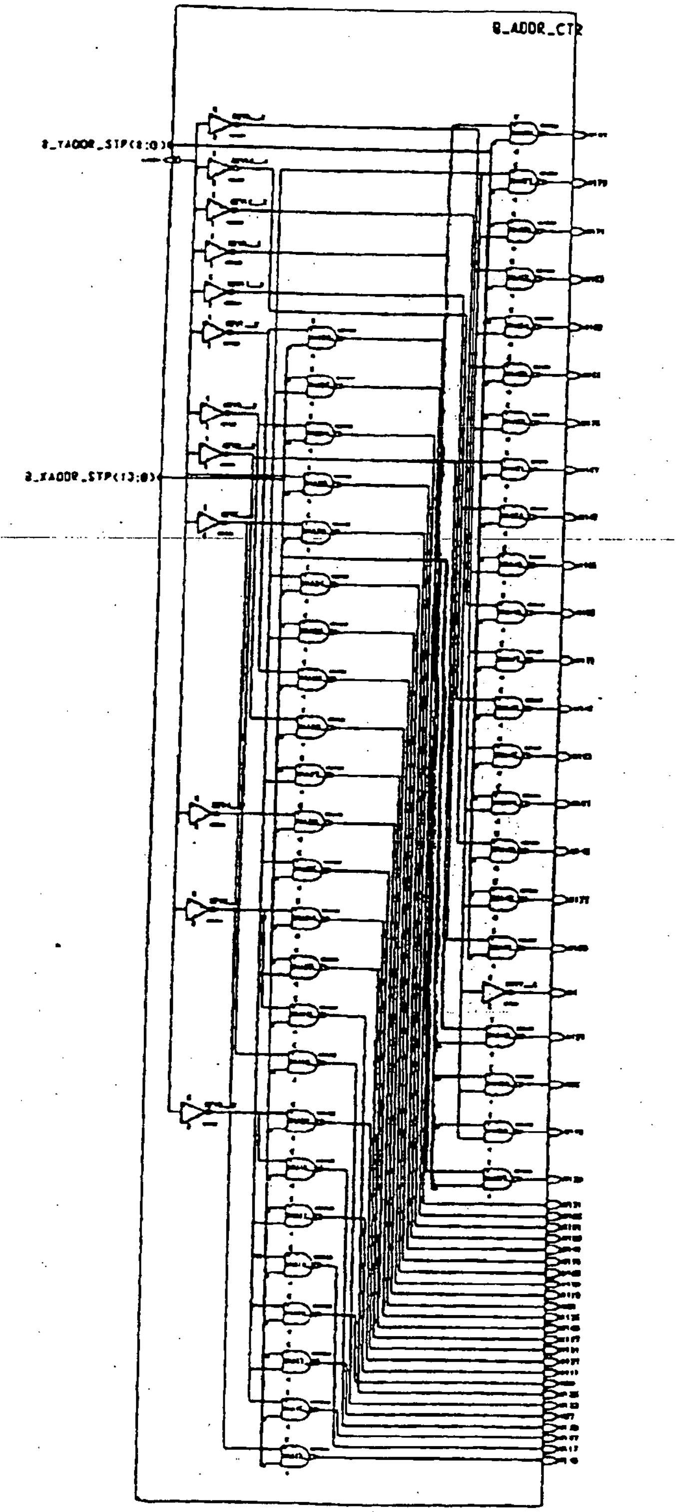


FIG. 10

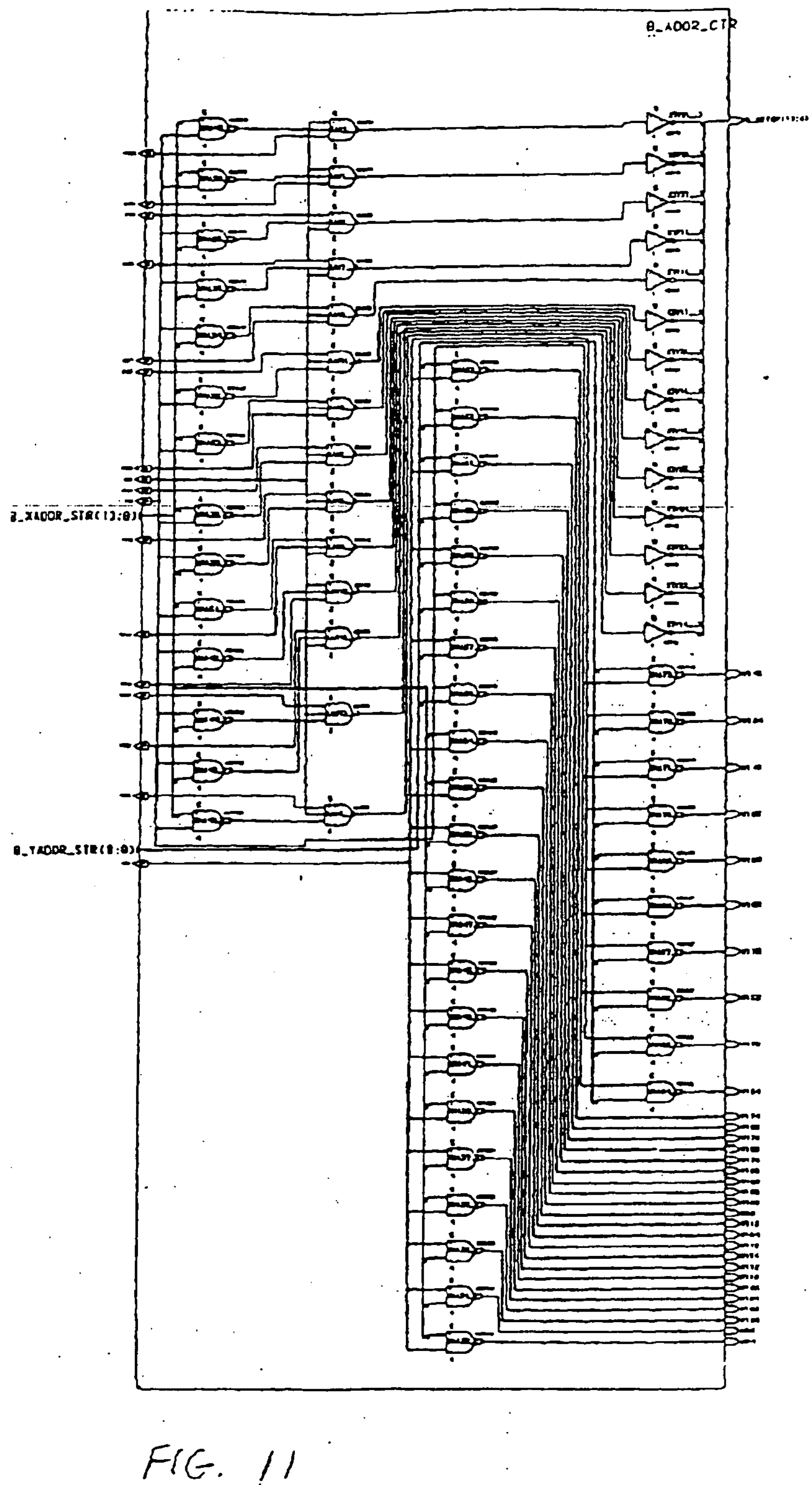


FIG. 11

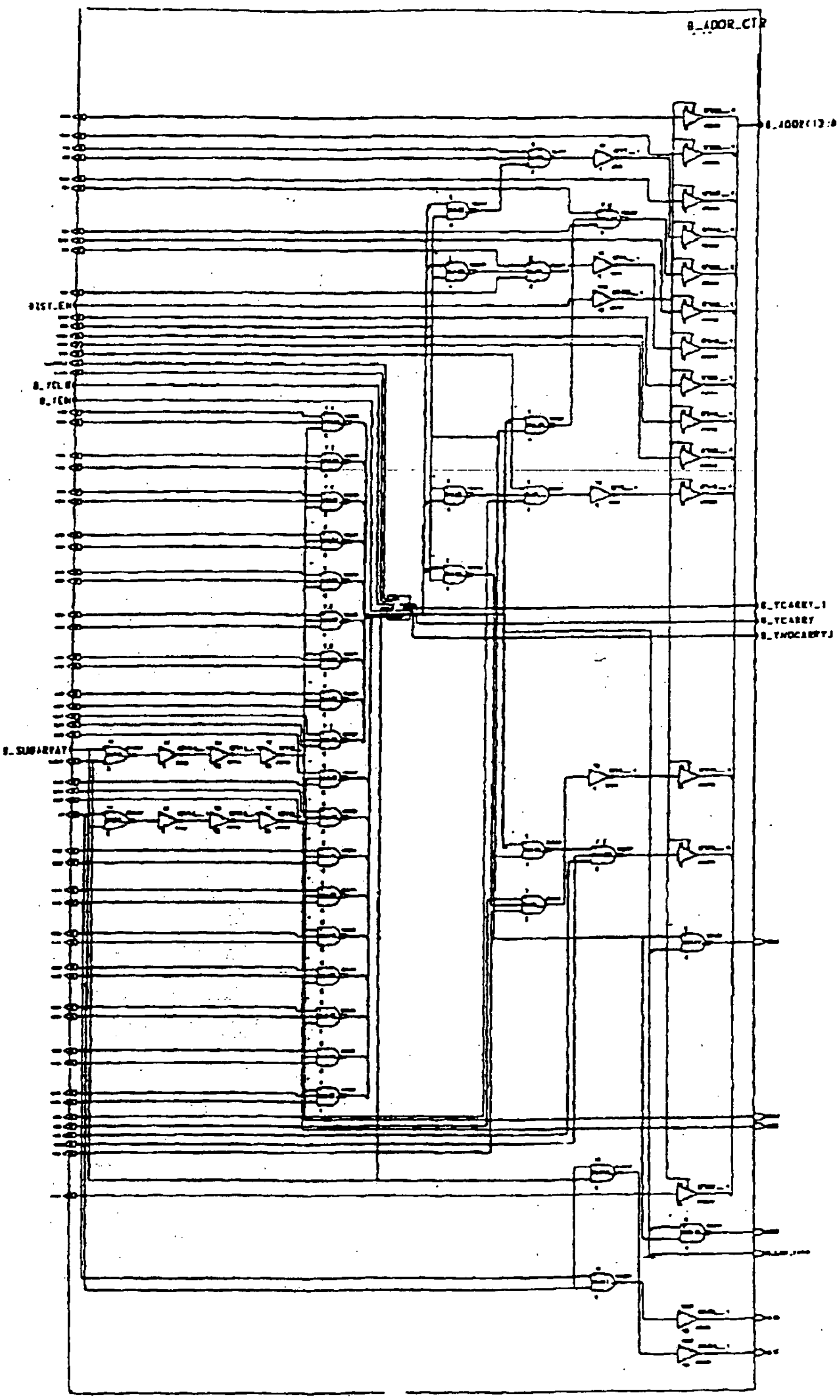


FIG. 12

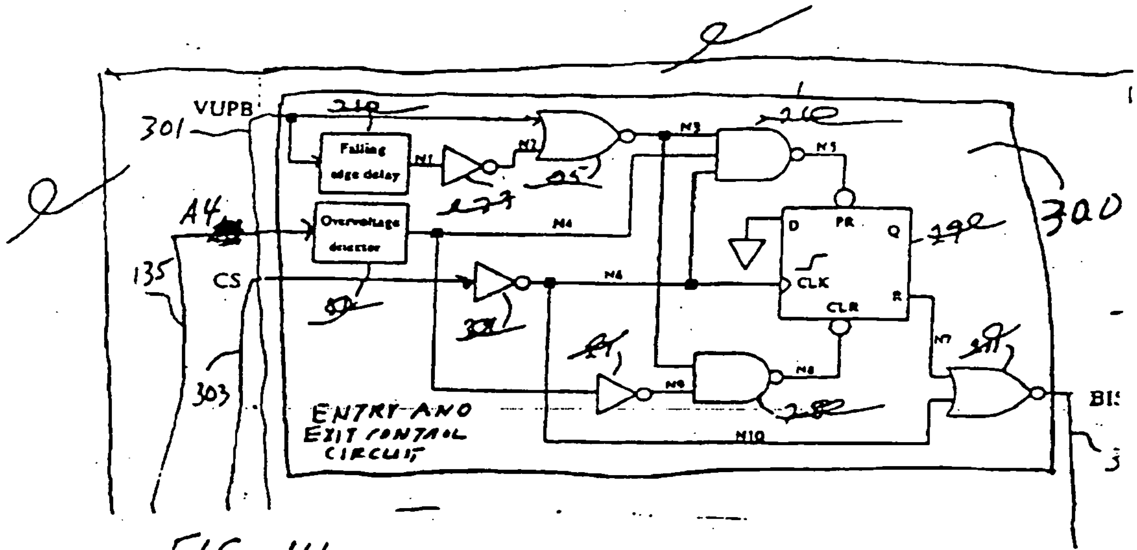


FIG. 14

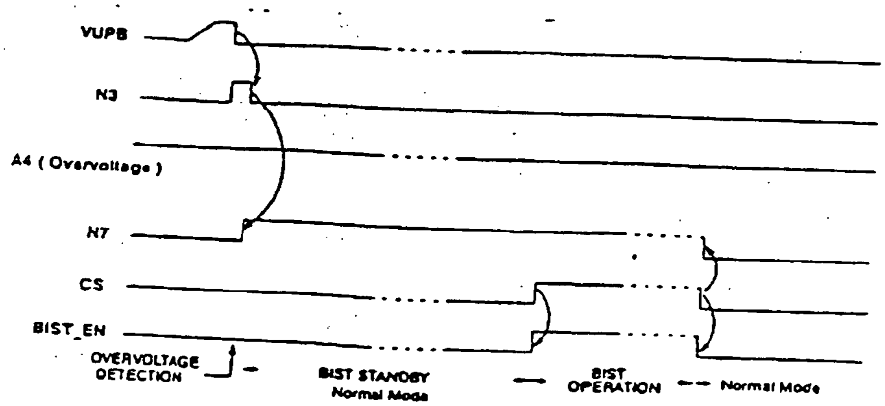


Figure 15

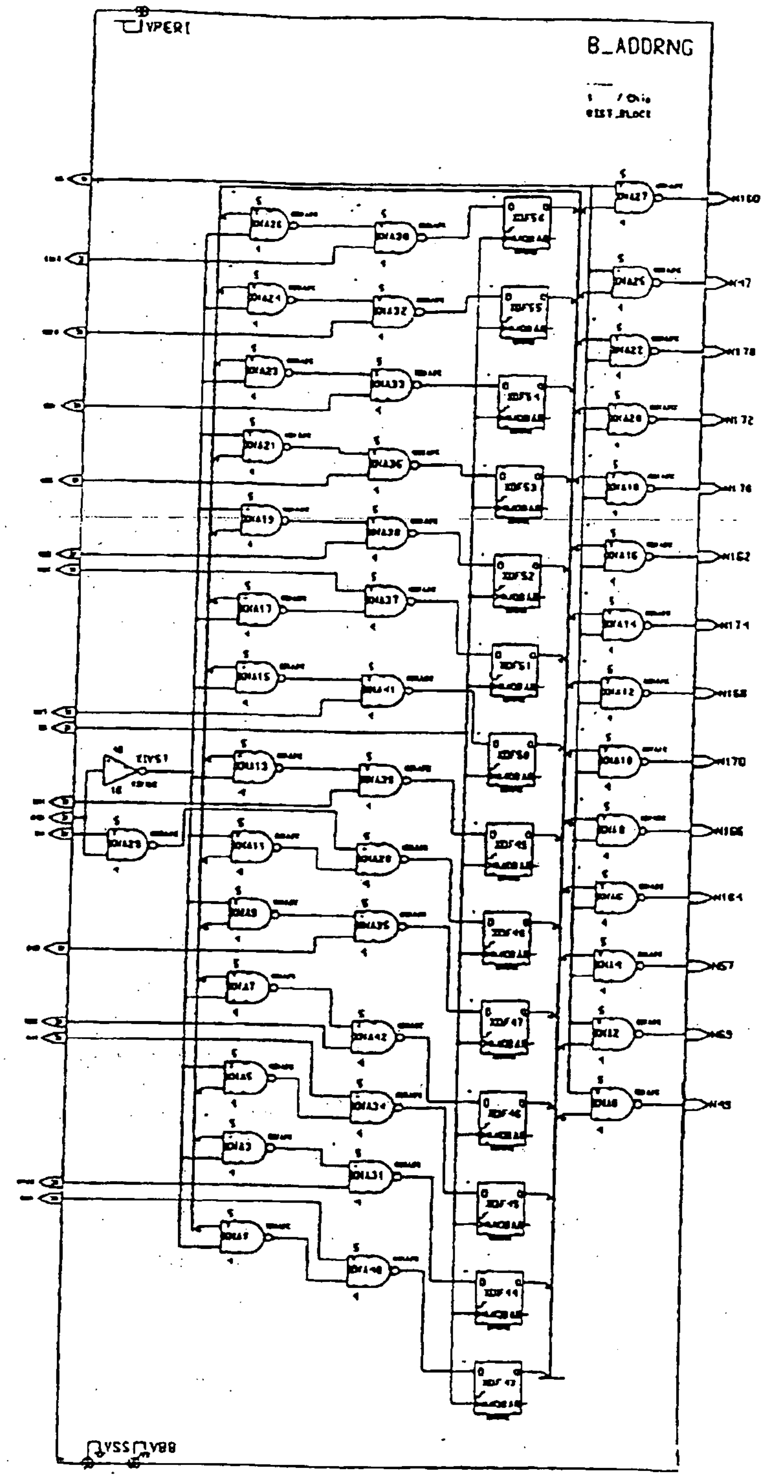


FIG. 16

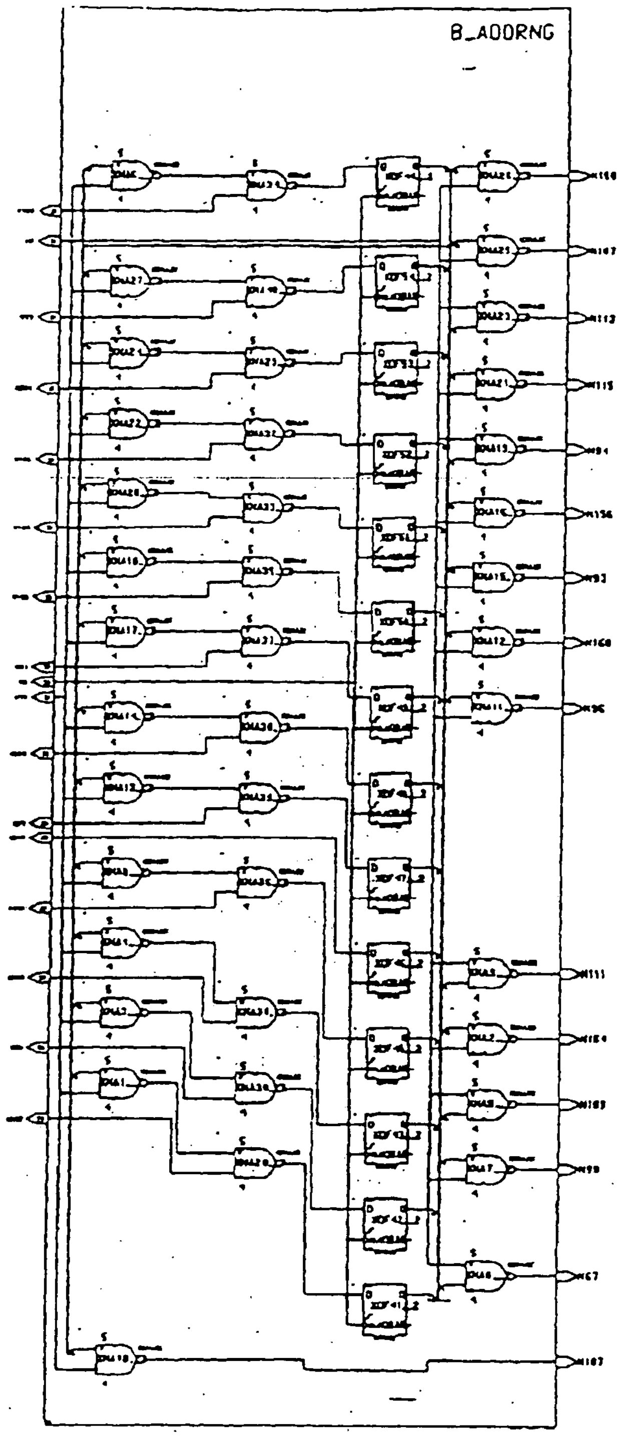


FIG. 17

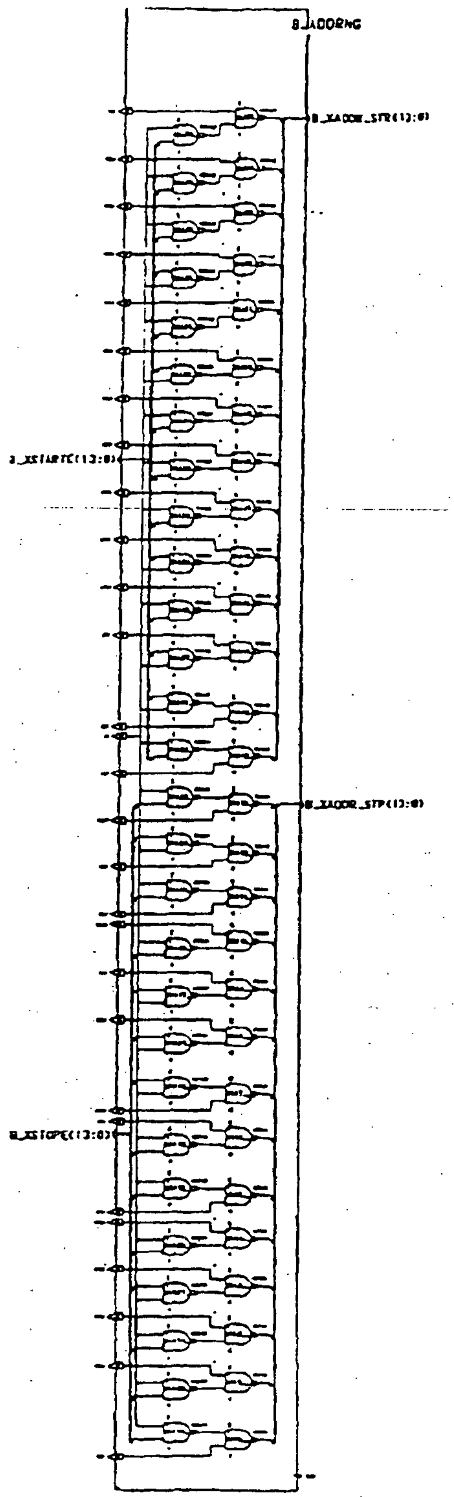
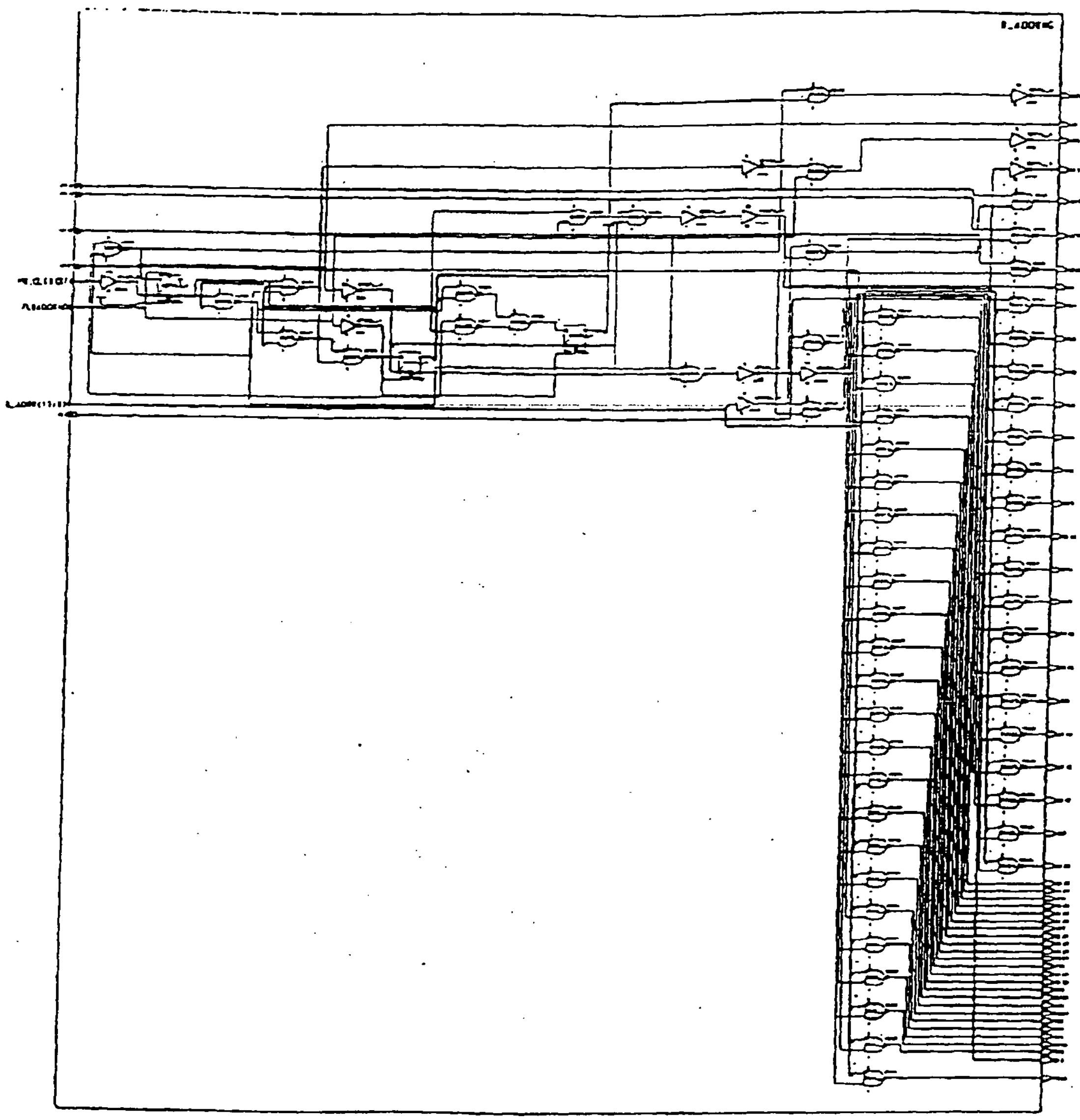

B_ADDRNG
B_XADDR_STR(13:0)
B_XSTARTE(13:0)
B_XADDR_STP(13:0)
B_XSTOPE(13:0)

FIG. 18

TI 2



TI 23011

FIG. 19

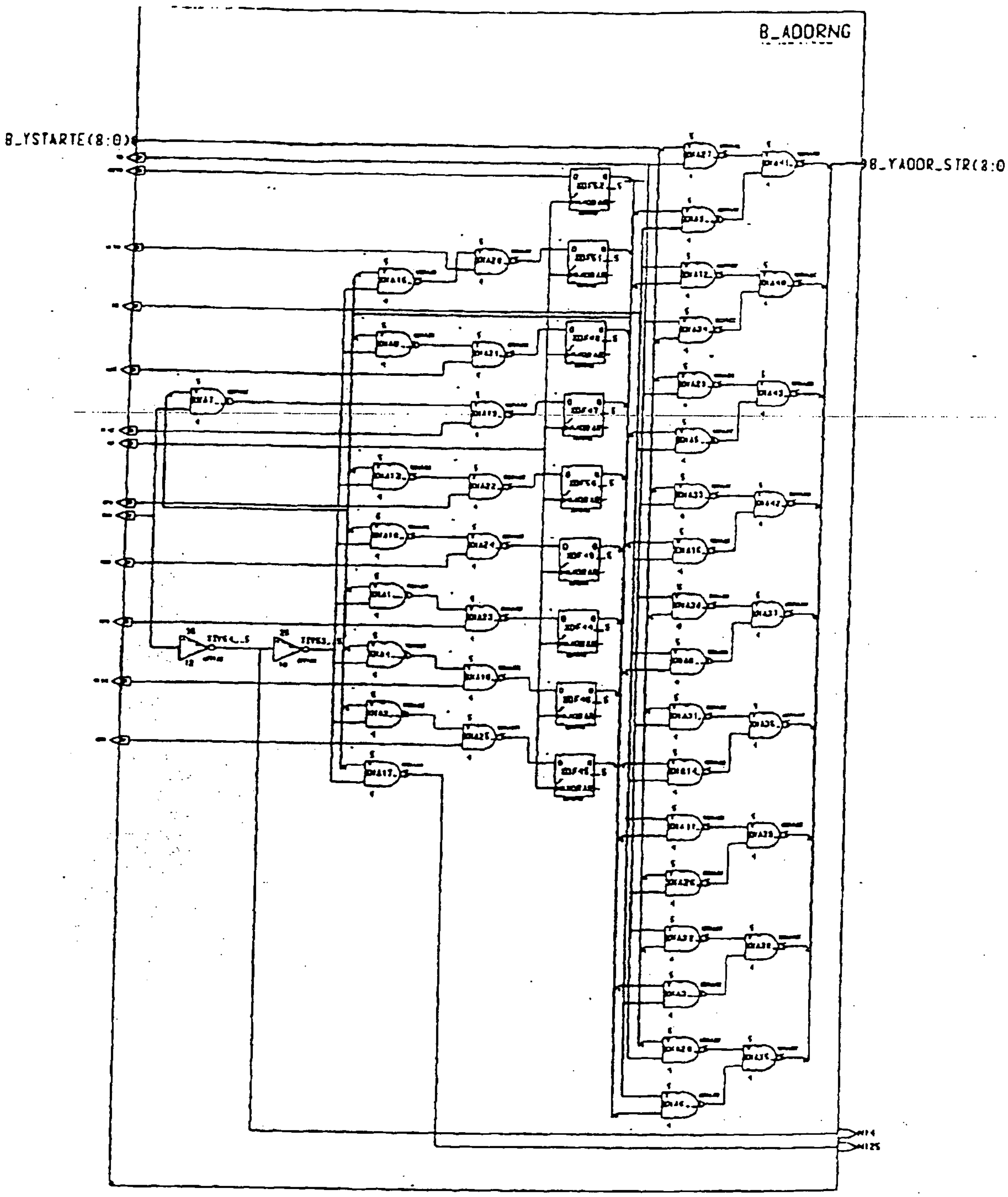


FIG. 20

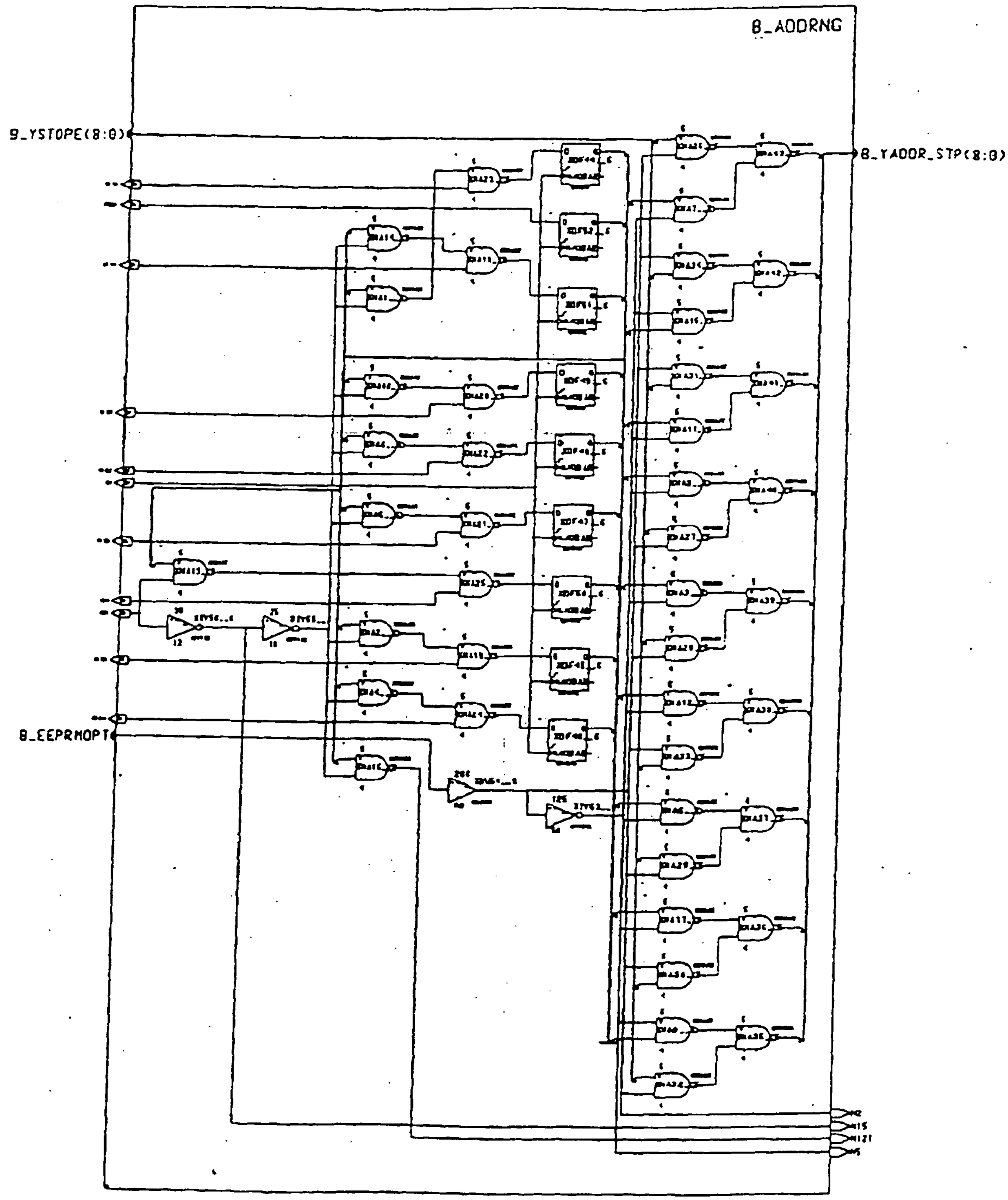


FIG. 21

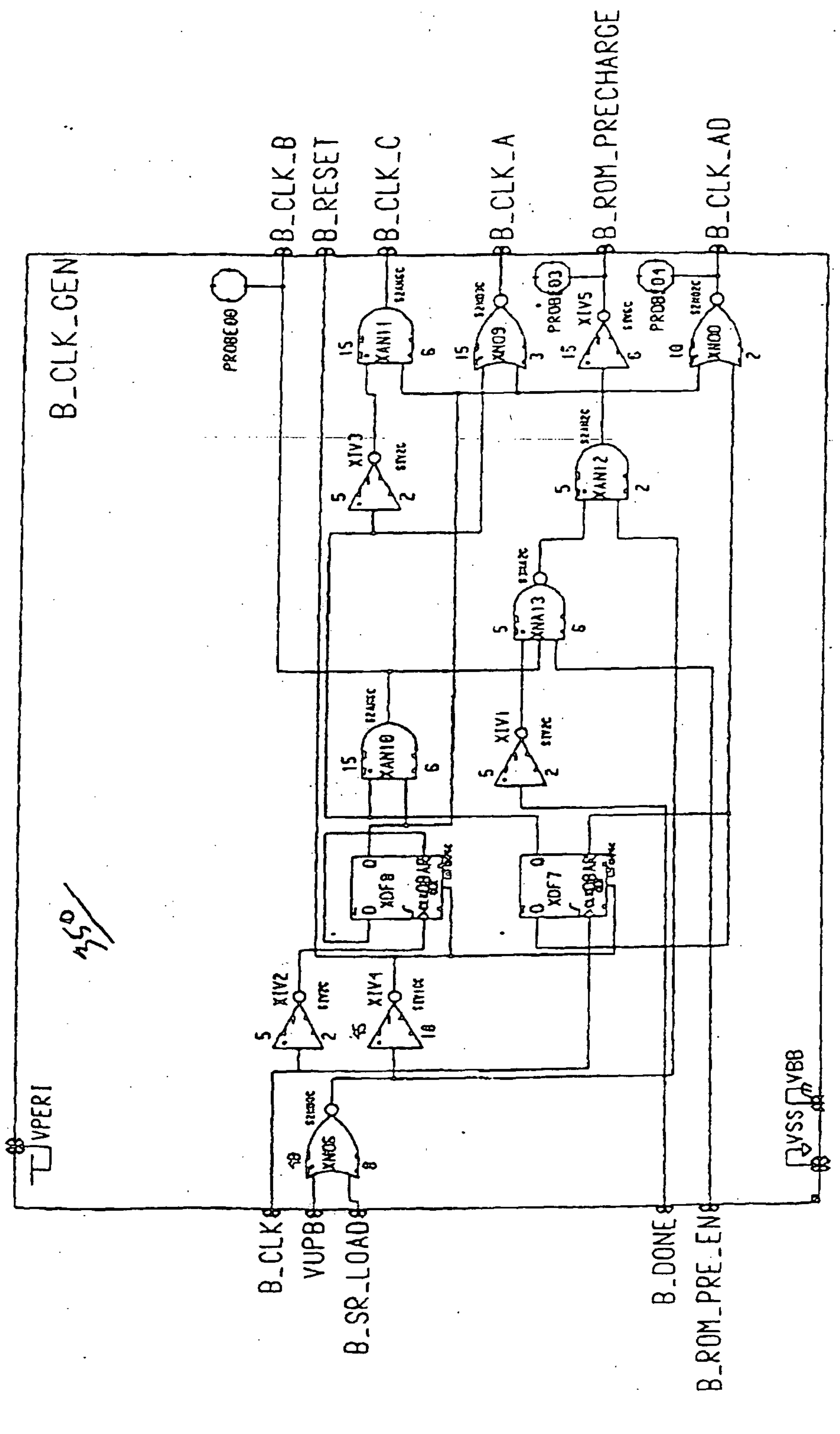

B_CLK_GEN
B_CLK_B
B_RESET
B_CLK_C
B_CLK_A
B_ROM_PRECHARGE
B_CLK_AD
B_CLK
VUPB
B_SR_LOAD
B_DONE
B_ROM_PRE_EN
VPERI
VSS
VBB

FIG. 22

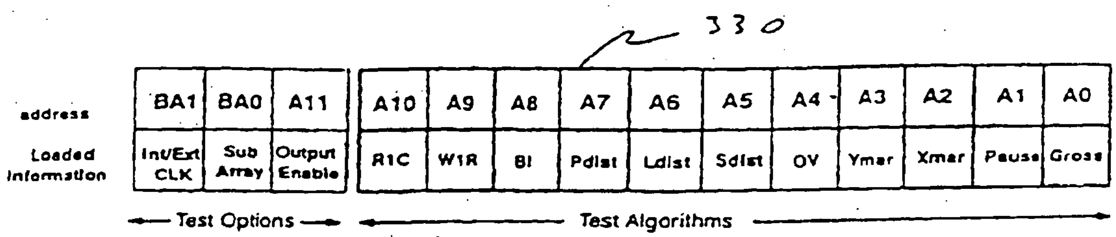

330
address
BA1
BA0
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
Loaded Information
Int/Ext CLK
Sub Array
Output Enable
R1C
W1R
BI
Pdist
Ldist
Sdist
OV
Ymar
Xmar
Pause
Gross
Test Options
Test Algorithms

FIG. 23

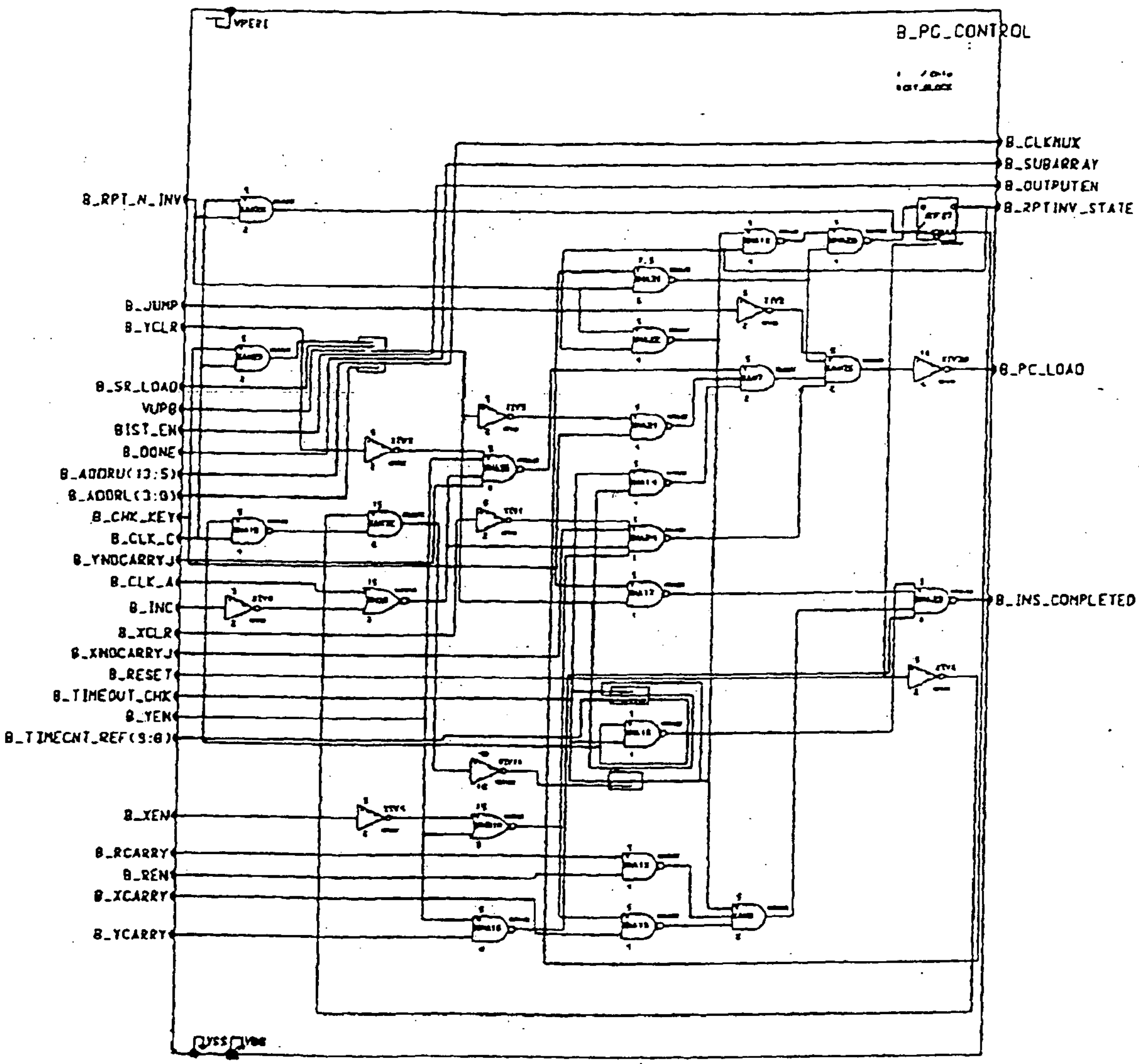

B_PC_CONTROL
B_RPT_N_INV
B_JUMP
B_YCLR
B_SR_LOAD
VUPB
BIST_EN
B_DONE
B_ADDRU(13:5)
B_ADDRL(3:0)
B_CHK_KEY
B_CLK_C
B_YNOCARRYJ
B_CLK_A
B_INC
B_XCLR
B_XNOCARRYJ
B_RESET
B_TIMEOUT_CHK
B_YEN
B_TIMECNT_REF(3:0)
B_XEN
B_RCARRY
B_REN
B_XCARRY
B_YCARRY
B_CLKMUX
B_SUBARRAY
B_OUTPUTEN
B_RPTINV_STATE
B_PC_LOAD
B_INS_COMPLETED
VSS
VDD

FIG. 24

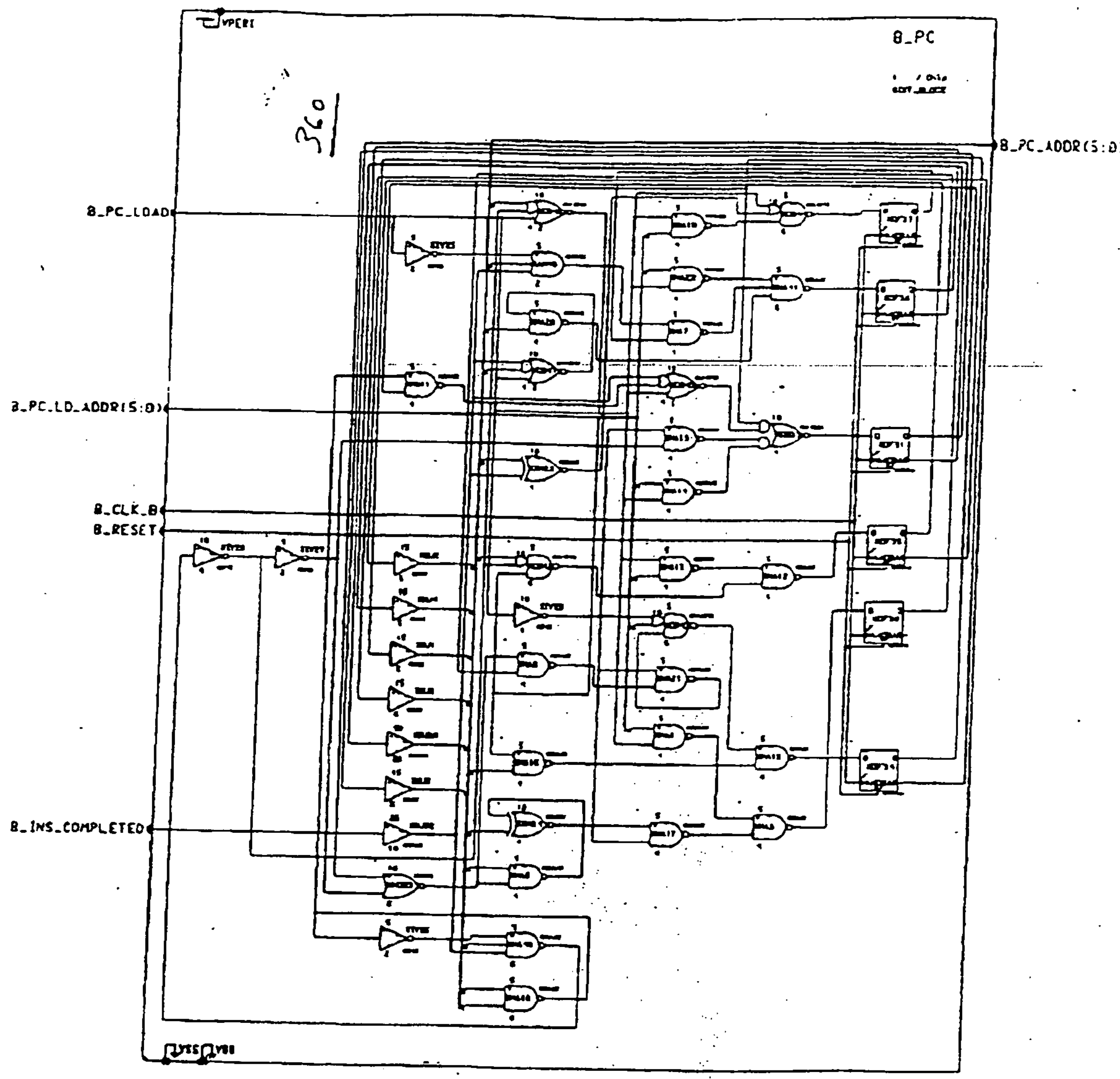

B_PC
B_PC_ADDR(5:0)
B_PC_LOAD
B_PC_LD_ADDR(5:0)
B_CLK_B
B_RESET
B_INS_COMPLETED
360

FIG. 25

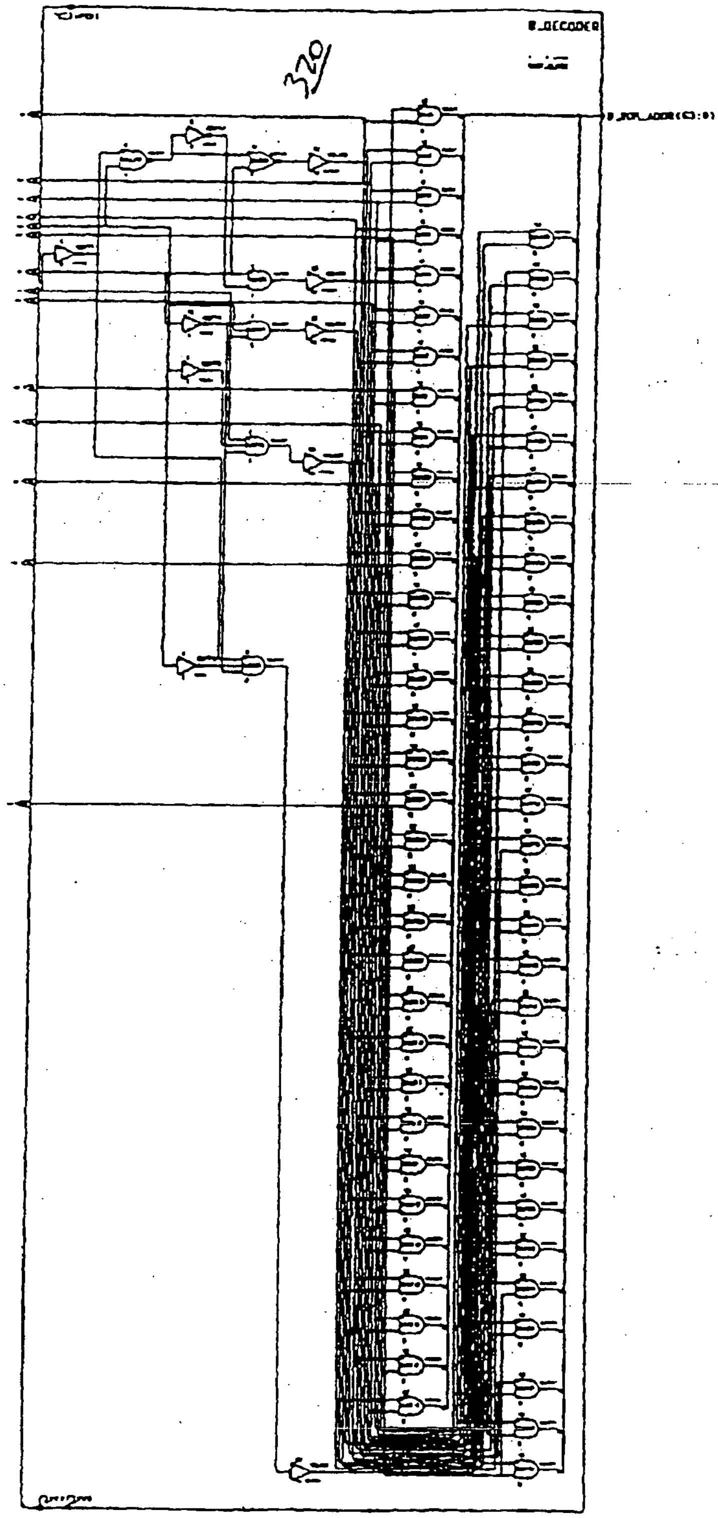


FIG. 26

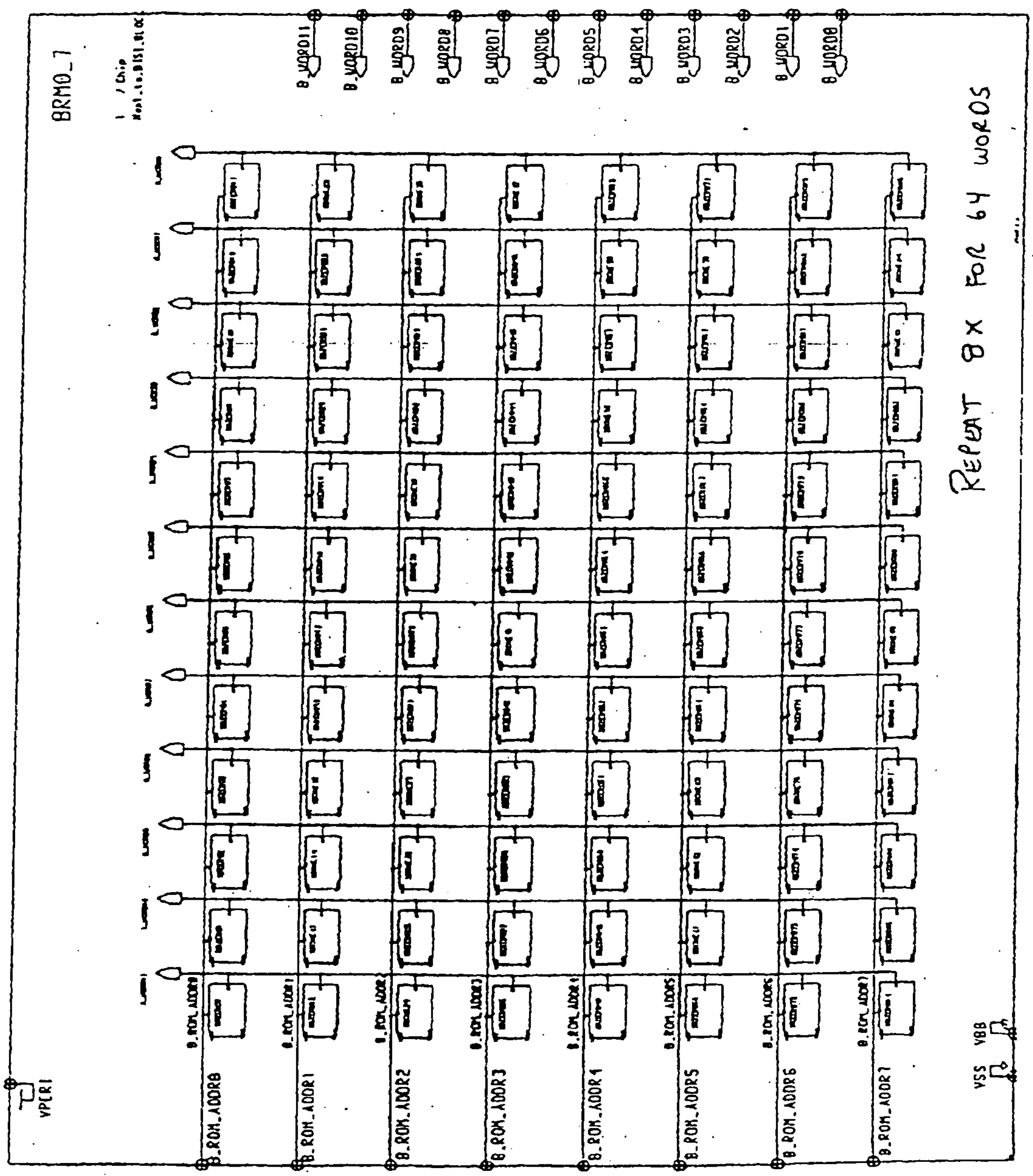
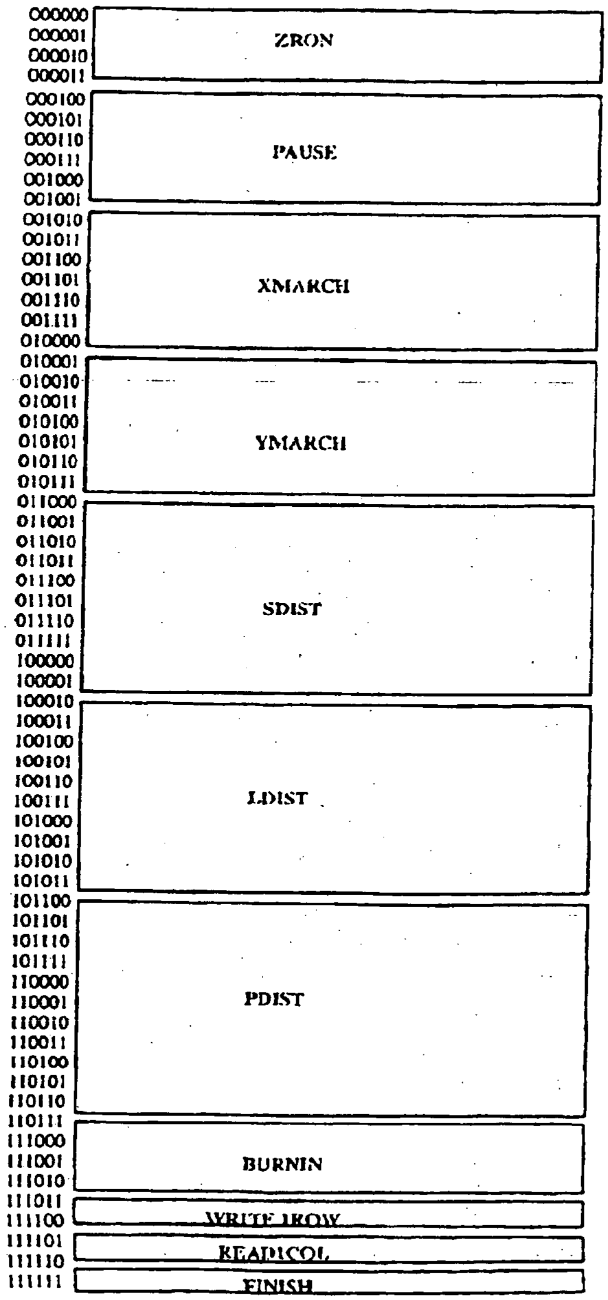

BRM0_7
B_WORD11
B_WORD10
B_WORD9
B_WORD8
B_WORD7
B_WORD6
B_WORD5
B_WORD4
B_WORD3
B_WORD2
B_WORD1
B_WORD0
REPEAT 8X FOR 64 WORDS
VPERI
B_ROM_ADDR0
B_ROM_ADDR1
B_ROM_ADDR2
B_ROM_ADDR3
B_ROM_ADDR4
B_ROM_ADDR5
B_ROM_ADDR6
B_ROM_ADDR7
VSS
VBB

FIG. 27

256M BIST ALGORITHM STORAGE CONFIGURATION



2

FIG 28

101100110001

Read	Write	X	Y	-		Tset0	Tset1	Data	ED	Ckbd	Alt	*1
				Test1	Test2	D5	D4	D3	D2	D1	D0	*2

Test1	Test2	Comments
0	0	Test if the test is enabled in enabled tests register 330.
0	1	Test for inverted pattern
1	0	Increment address by 1, and check for overflow (array address)
1	1	Increment counter and check if required value is reached (timing counter)

Tset0	Tset1	Comments
0	0	Timing set A
0	1	Timing set C
1	0	Timing set B
1	1	Timing set A, decrement

*1 Memory Access Instruction

*2 Program Control Instruction

FIG. 29

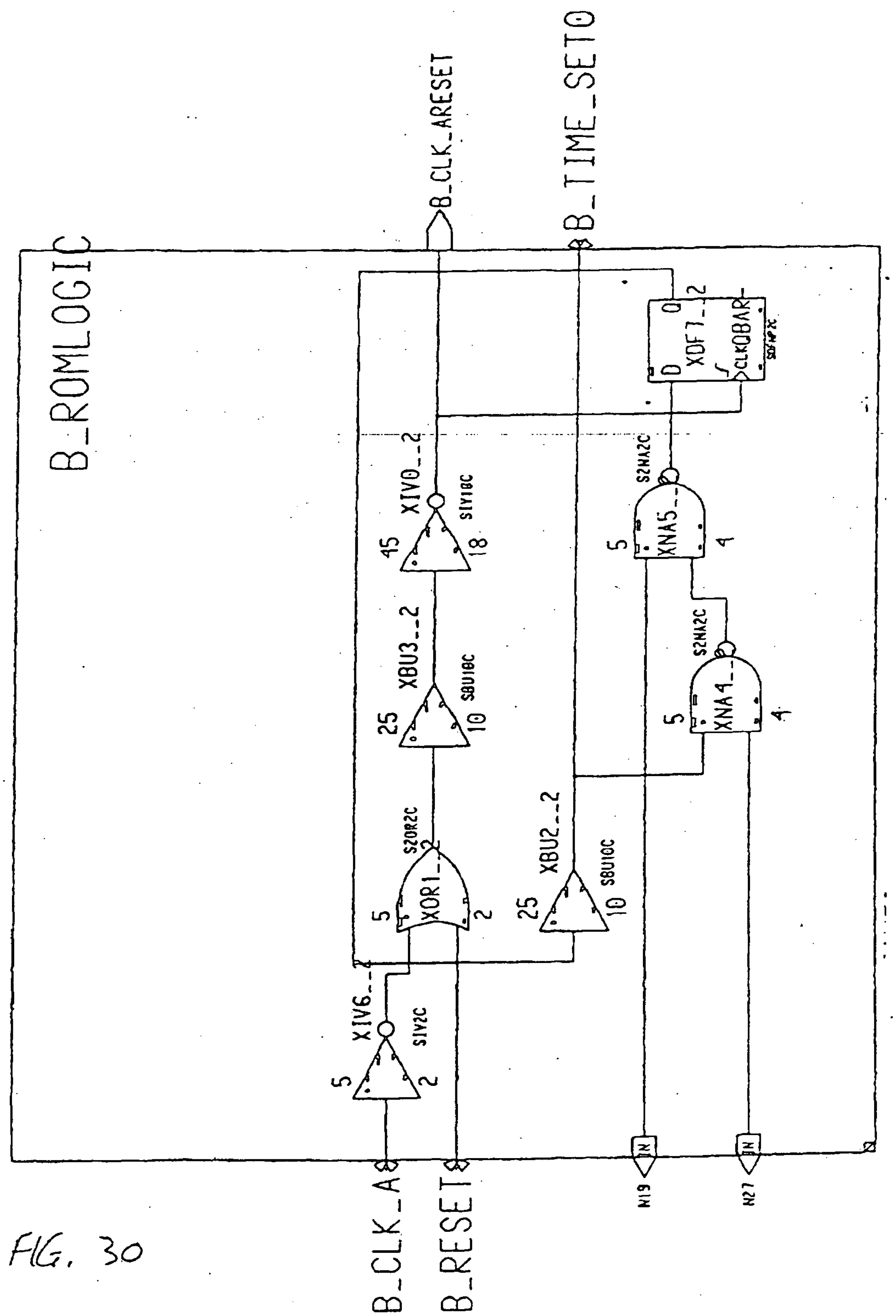

B_ROMLOGIC
B_CLK_ARESET
B_TIME_SETO
XDF7__2
XIV0__2
XBU3__2
XOR1__2
XIV6__
XBU2__2
XNA5__
XNA4__
B_CLK_A
B_RESET
N19
N27

FIG. 30

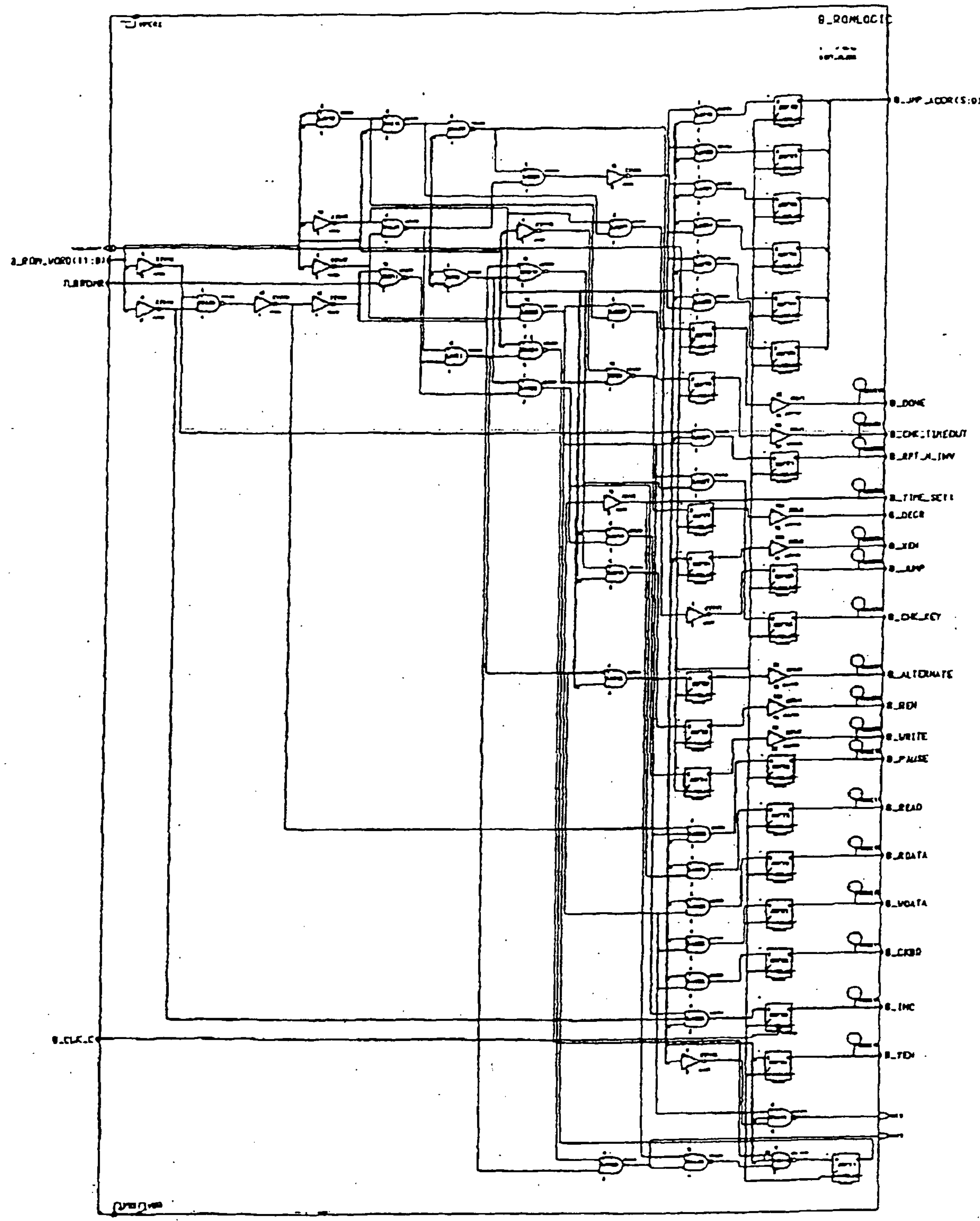

B_ROMLOGIC
B_ROM_WORD(11:0)
B_CLK_C
B_DONE
B_CHK_TIMEOUT
B_TIME_SET1
B_DECR
B_JMP
B_CHK_KEY
B_ALTERNATE
B_WRITE
B_PAUSE
B_READ
B_RDATA
B_WDATA
B_INC

FIG. 31

```
architecture behav of b_passfail is
signal B_EXPECT_DATA   : std_logic_vector (3 downto 0 );
signal B_EXPECT_DATAB  : std_logic_vector (3 downto 0 );
signal B_WRITE_DATA    : std_logic_vector (3 downto 0 );
signal CKBDI, TRUEWRITE, B_REVNVAL, B_RODDVAL, B_WEVNVAL, B_WODDVAL : std_logic;

begin

 CKBDI <= B_CKBD and ( B_LSB_XADD xor B_LSB_YADD);

 pass_fail_check : process(B_CLK, B_OEB, B_RESET)

              begin
                if (B_RESET = '1' ) then
                  B_PF_RESULT <= '0';
                elsif (B_CLK'EVENT and B_CLK = '1') then
                   if (B_OEB = '0' and (B_DQ(31 downto 28) /= B_EXPECT_DATA
                        or B_DQ(27 downto 24) /= B_EXPECT_DATAB)) then
                      B_PF_RESULT <= '1';
                   end if;
                end if;
              end process pass_fail_check;

   B_REVNVAL <= B_RDATA xor CKBDI;
   B_RODDVAL <= B_ALTERNATE xor B_RDATA xor CKBDI;
   B_WEVNVAL <= B_WDATA xor CKBDI;
   B_WODDVAL <= B_ALTERNATE xor B_WDATA xor CKBDI;

   B_EXPECT_DATA(0) <= B_REVNVAL;
   B_EXPECT_DATA(1) <= B_RODDVAL;
   B_EXPECT_DATA(2) <= B_REVNVAL;
   B_EXPECT_DATA(3) <= B_RODDVAL;
   B_EXPECT_DATAB    <= not B_EXPECT_DATA;

   B_WRITE_DATA(0) <= B_WEVNVAL;
   B_WRITE_DATA(1) <= B_WODDVAL;
   B_WRITE_DATA(2) <= B_WEVNVAL;
   B_WRITE_DATA(3) <= B_WODDVAL;

  monitor_data_values : process(B_WRITE, B_REVNVAL, B_RODDVAL, B_WEVNVAL, B_WODDVAL)
     begin
       if ( B_WRITE = '1') then
         B_DEVEN <= B_WEVNVAL;
         B_DODD  <= B_WODDVAL;
       else
         B_DEVEN <= B_REVNVAL;
         B_DODD  <= B_RODDVAL;
       end if;
     end process monitor_data_values;

   TRUEWRITE <= B_WRITE and (not B_WDATA_TIMB) and (not TLBMON) and
                (not TLBROMR);

   guarded_block :   block ( TRUEWRITE ='1' )
   begin
       B_DQ(31 downto 28) <= guarded B_WRITE_DATA;
       B_DQ(27 downto 24) <= guarded B_WRITE_DATA;
   end block guarded_block;

end behav;
```

FIG. 32

FIG. 33

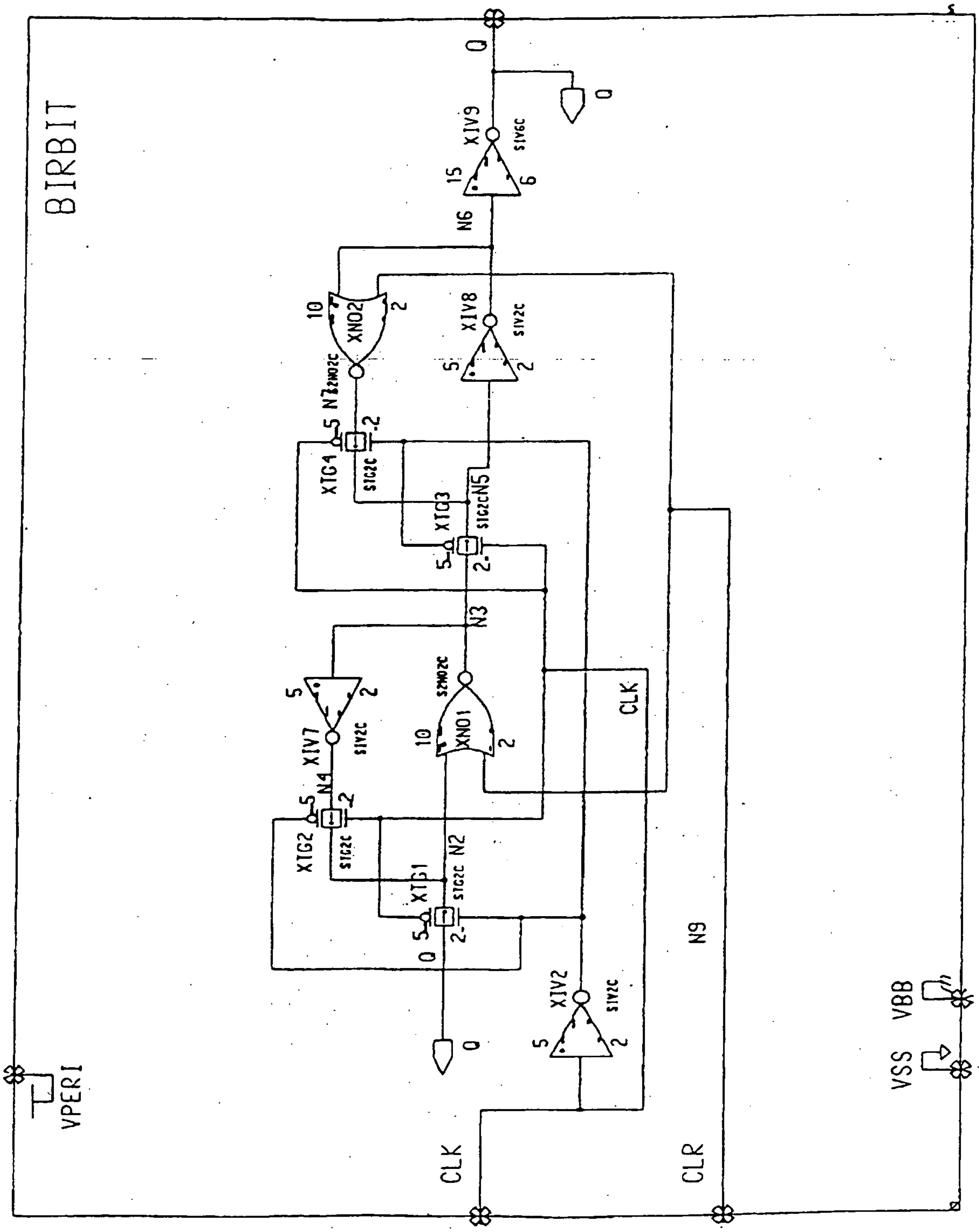
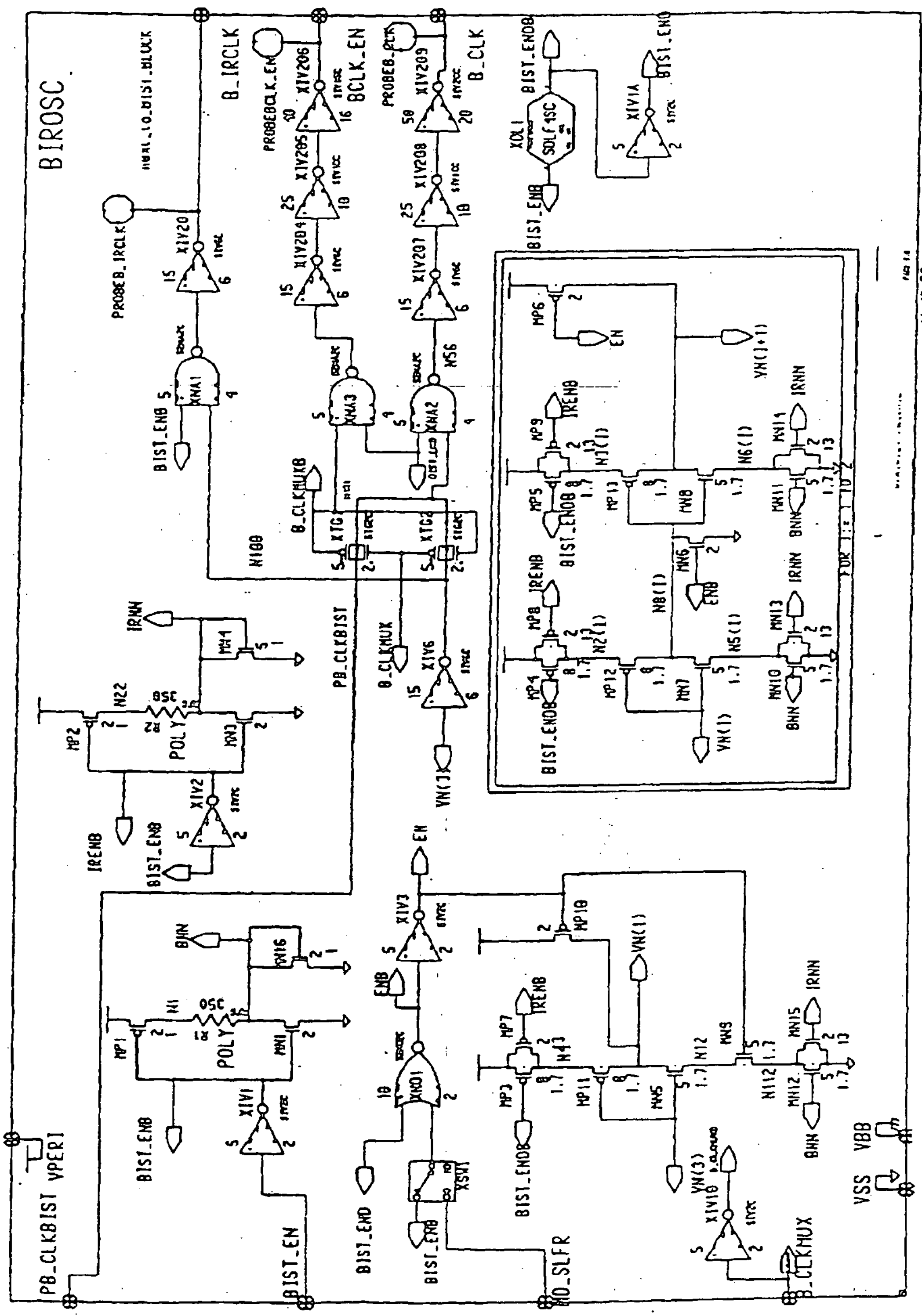

BIRBIT
XIV9
XIV8
XIV7
XIV2
XNO1
XNO2
XTG1
XTG2
XTG3
XTG4
N2
N3
N4
N5
N6
N7
N9
Q
CLK
CLR
VPERI
VSS
VBB

FIG. 34


BIROSC
B_IRCLK
PROBEB_IRCLK
BCLK_EN
PROBEBCLK_EN
B_CLK
BIST_ENOB
BIST_EN
BIST_ENB
B_CLKMUXB
B_CLKMUX
PB_CLKBIST
IRNN
IRENB
BNN
EN
ENB
VN(1)
VN(3)
POLY
PB_CLKBIST VPERI
IO_SLFR
VSS VBB
11:49:55 am

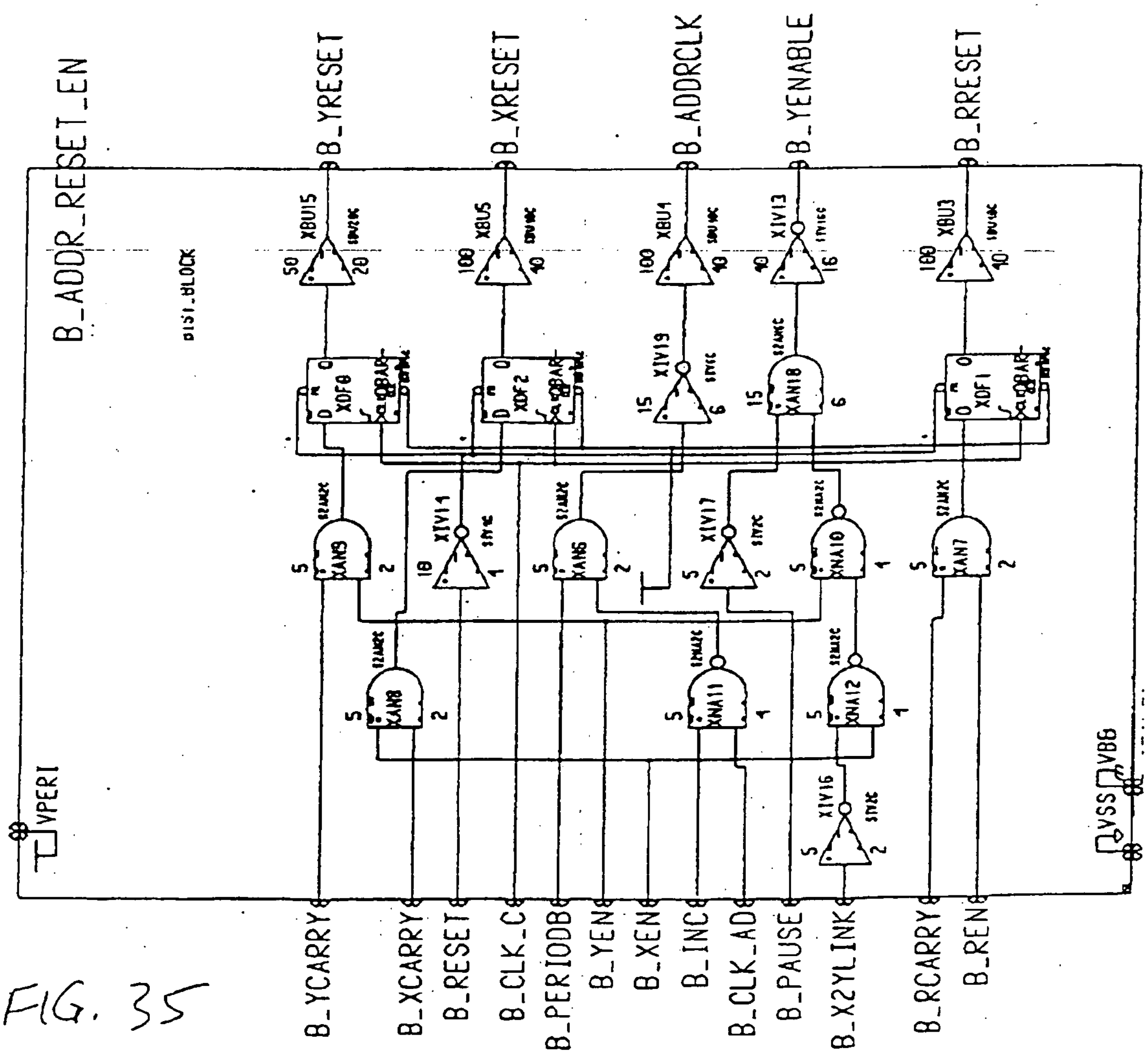

B_ADDR_RESET_EN
B_YRESET
B_XRESET
B_ADDRCLK
B_YENABLE
B_RRESET
XBU15
XBU5
XBU4
XIV13
XBU3
XDF0
XDF2
XIV19
XAN18
XDF1
XAN3
XIV14
XAN5
XIV17
XNA10
XAN7
XAN8
XNA11
XNA12
XIV16
VPERI
VSS
VBB
B_YCARRY
B_XCARRY
B_RESET
B_CLK_C
B_PERIODB
B_YEN
B_XEN
B_INC
B_CLK_AD
B_PAUSE
B_X2YLINK
B_RCARRY
B_REN

FIG. 35

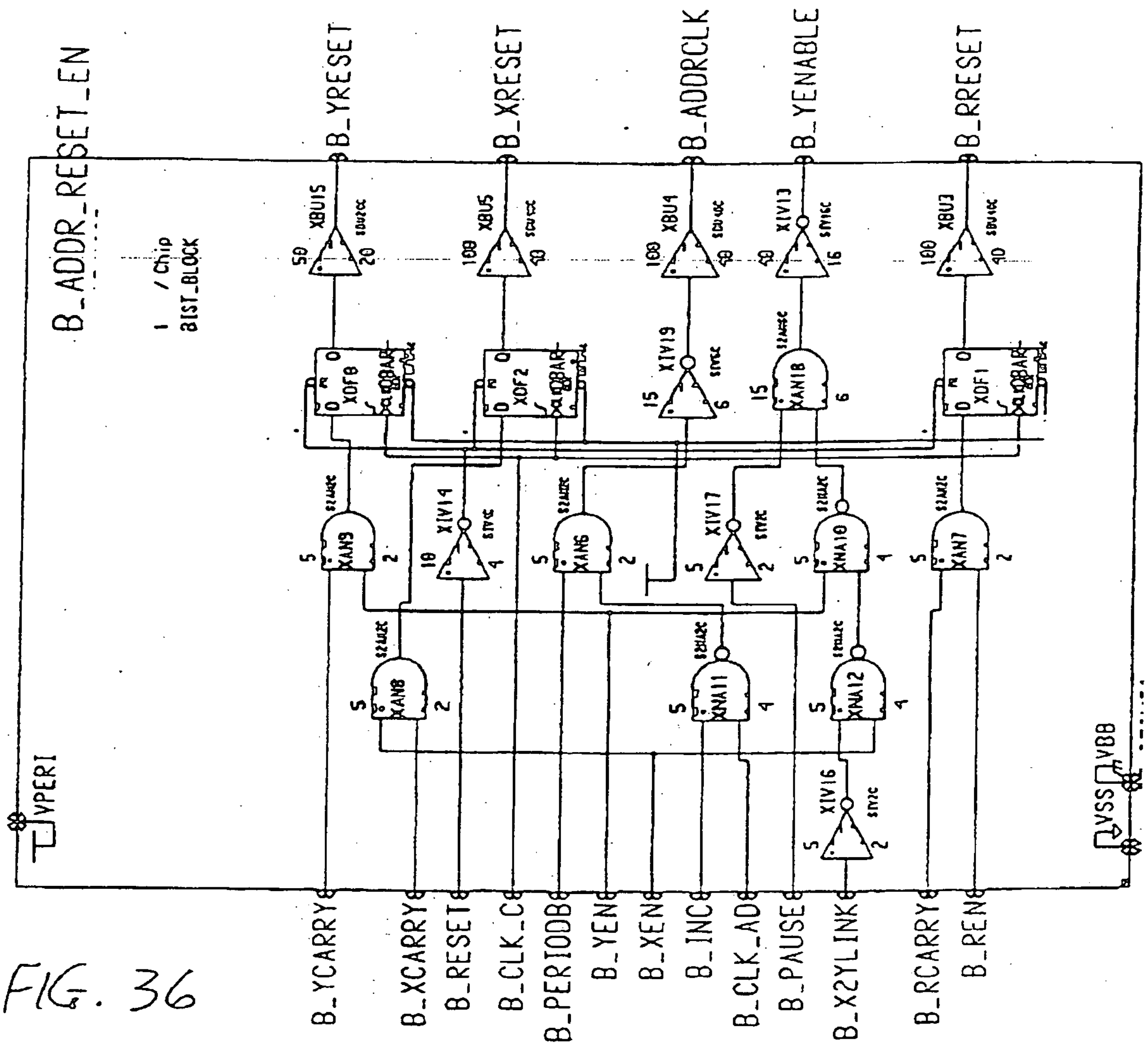

B_ADDR_RESET_EN
1 / Chip
BIST_BLOCK
B_YRESET
B_XRESET
B_ADDRCLK
B_YENABLE
B_RRESET
XBU15
XBU5
XBU4
XIV13
XBU3
XDF8
XDF2
XIV19
XAN18
XDF1
XAN9
XIV14
XAN6
XIV17
XNA10
XAN7
XAN8
XNA11
XNA12
XIV16
VPERI
VSS
VBB
B_YCARRY
B_XCARRY
B_RESET
B_CLK_C
B_PERIODB
B_YEN
B_XEN
B_INC
B_CLK_AD
B_PAUSE
B_X2YLINK
B_RCARRY
B_REN

FIG. 36

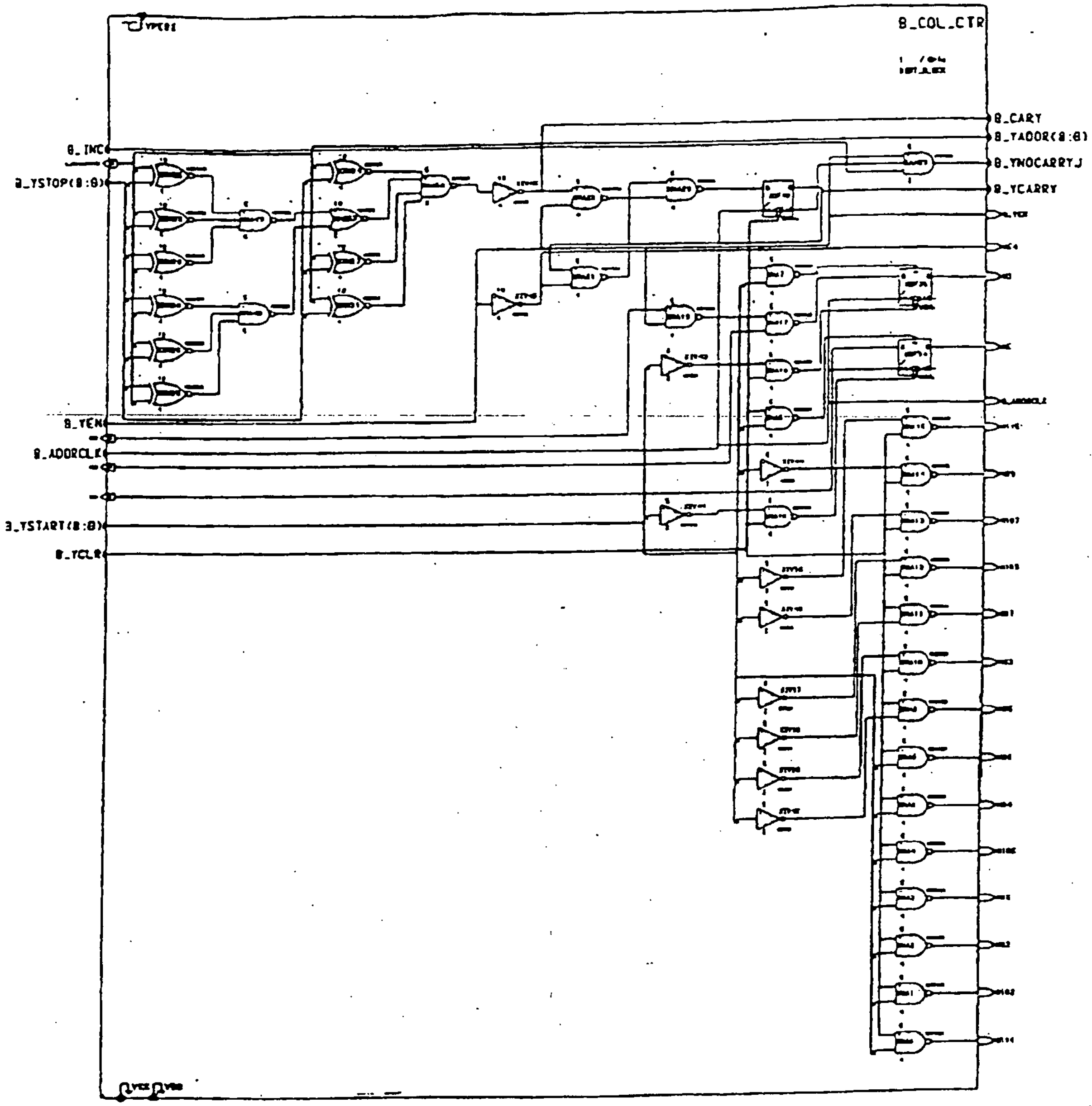

B_COL_CTR
B_INC
B_YSTOP(8:0)
B_YEN
B_ADDRCLK
B_YSTART(8:0)
B_YCLR
B_CARY
B_YADDR(8:0)
B_YNOCARRYJ
B_YCARRY

FIG. 37

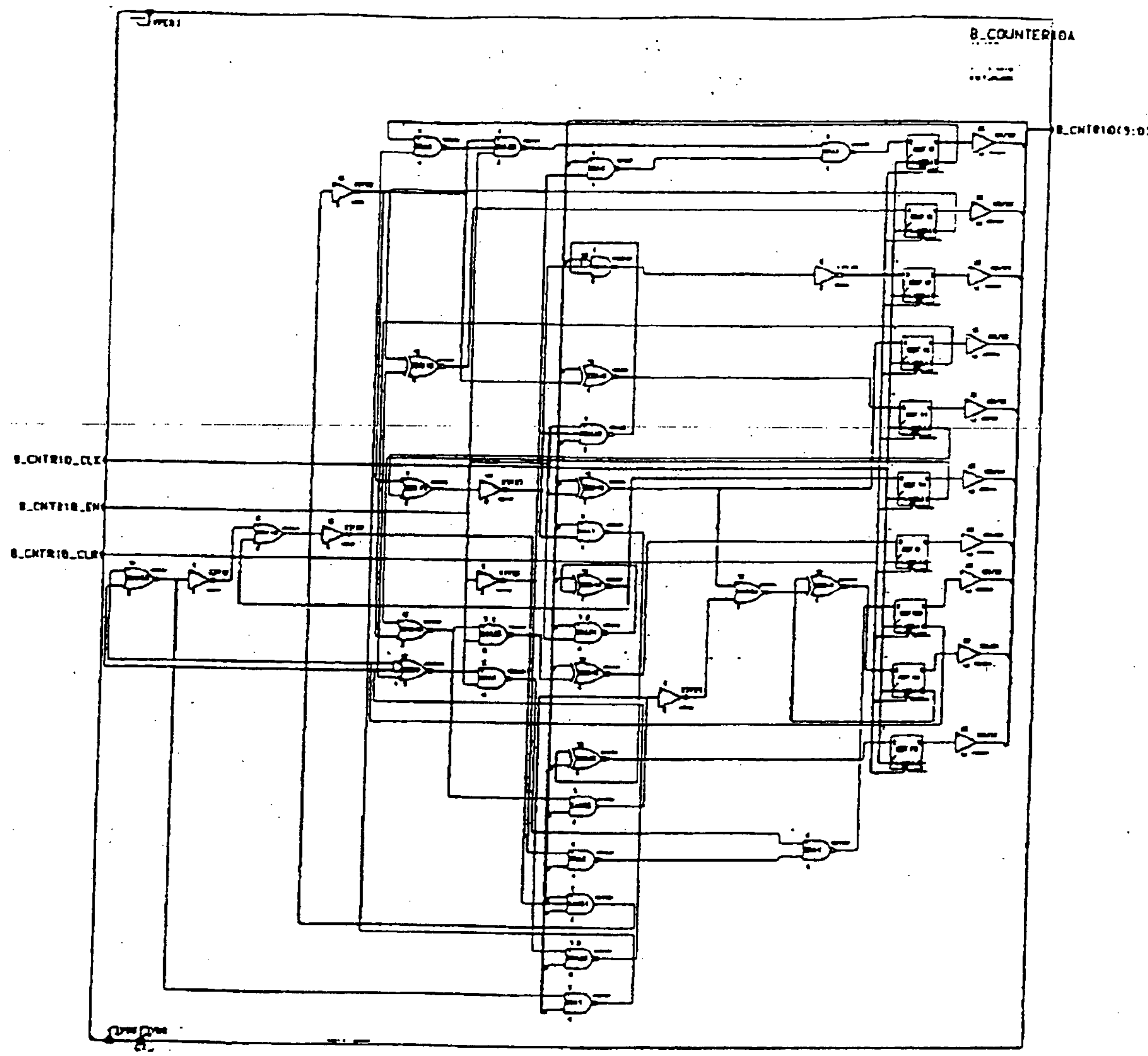


FIG. 38

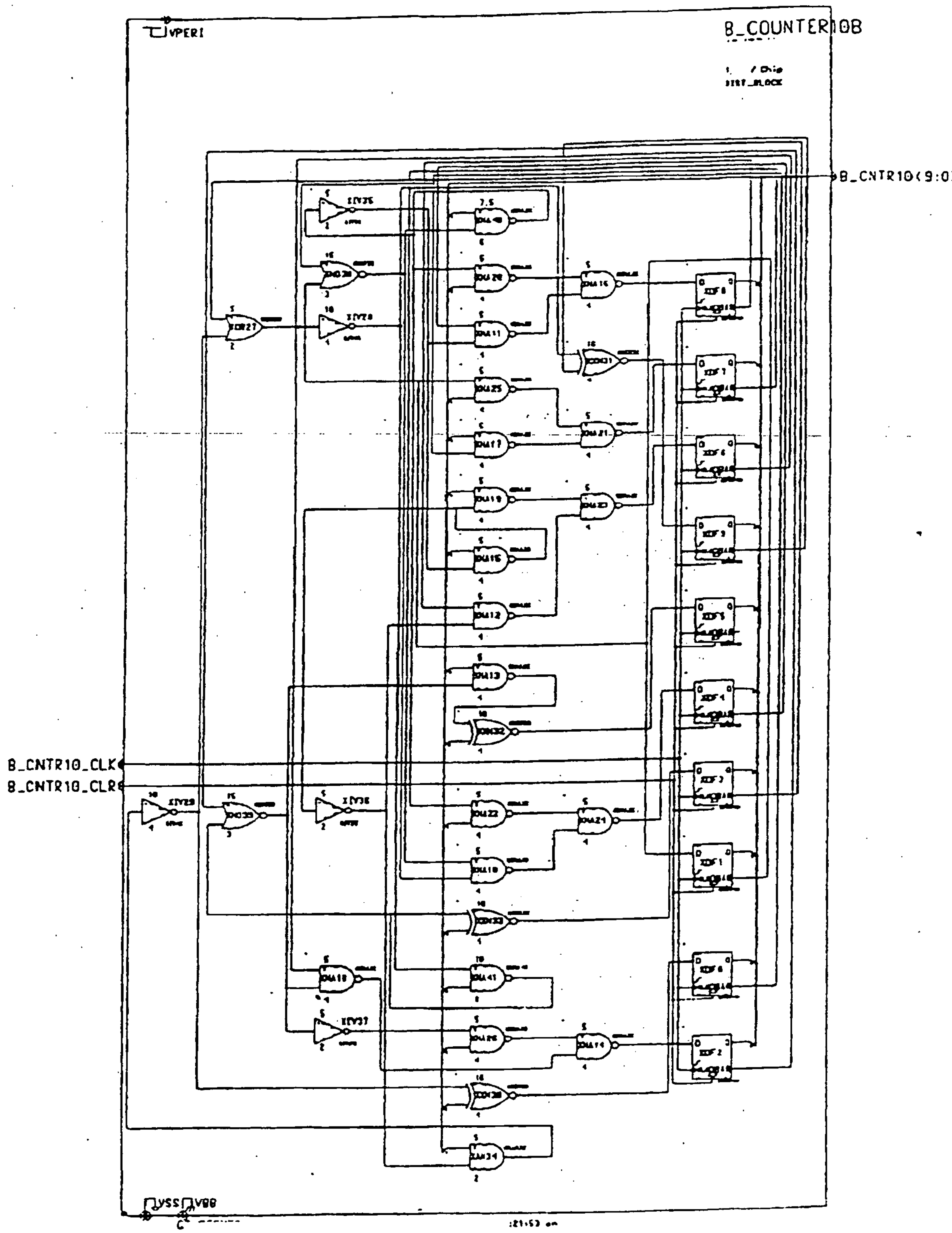

B_COUNTER10B
B_CNTR10(9:0)
B_CNTR10_CLK
B_CNTR10_CLR
VPERI
VSS
VBB

FIG. 39

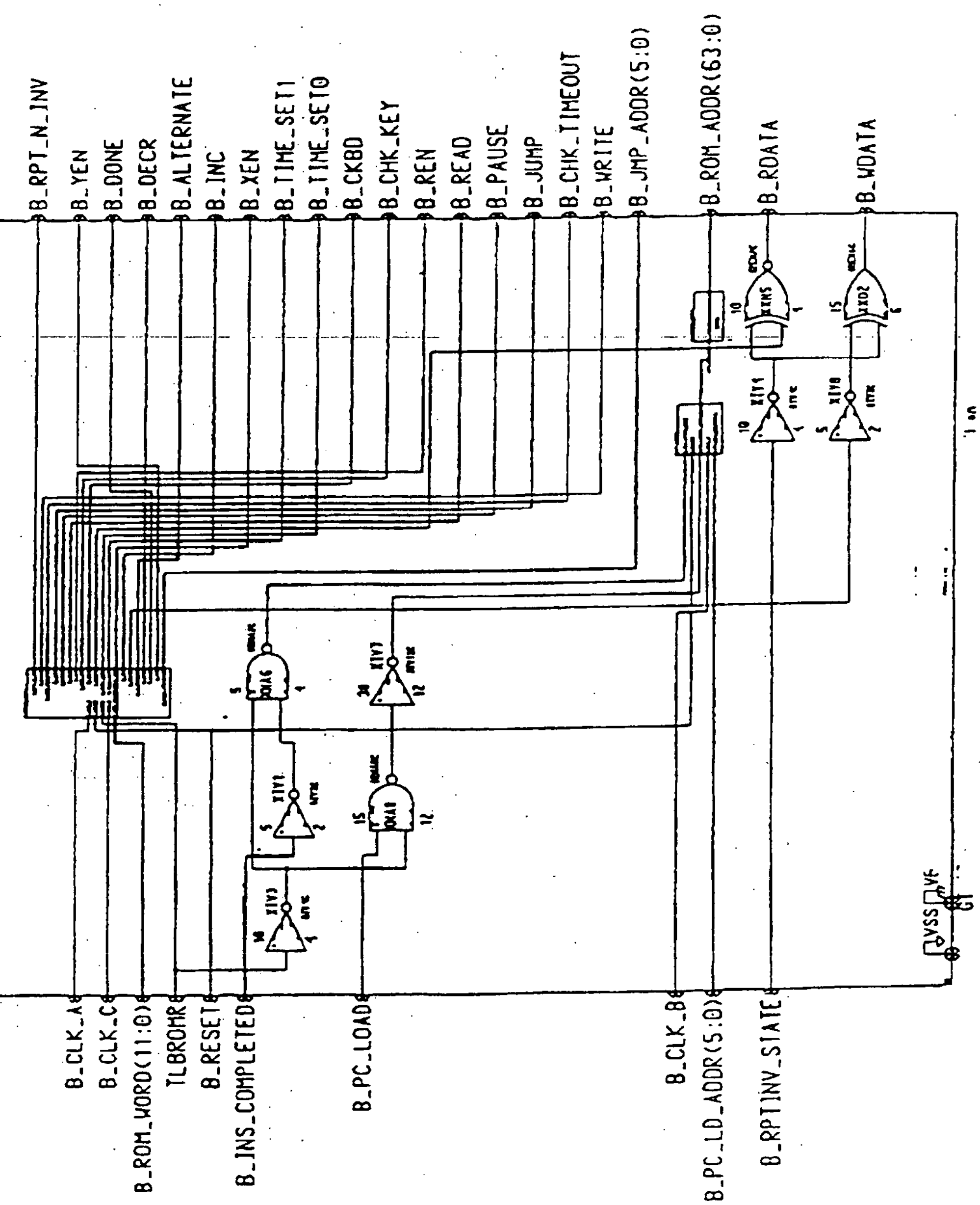

B_CROM
VPERI
B_RPT_N_INV
B_YEN
B_DONE
B_DECR
B_ALTERNATE
B_INC
B_XEN
B_TIME_SET1
B_TIME_SET0
B_CKBD
B_CHK_KEY
B_REN
B_READ
B_PAUSE
B_JUMP
B_CHK_TIMEOUT
B_WRITE
B_JMP_ADDR(5:0)
B_ROM_ADDR(63:0)
B_RDATA
B_WDATA
B_CLK_A
B_CLK_C
B_ROM_WORD(11:0)
TLBROMR
B_RESET
B_INS_COMPLETED
B_PC_LOAD
B_CLK_B
B_PC_LD_ADDR(5:0)
B_RPTINV_STATE
VSS

FIG. 40

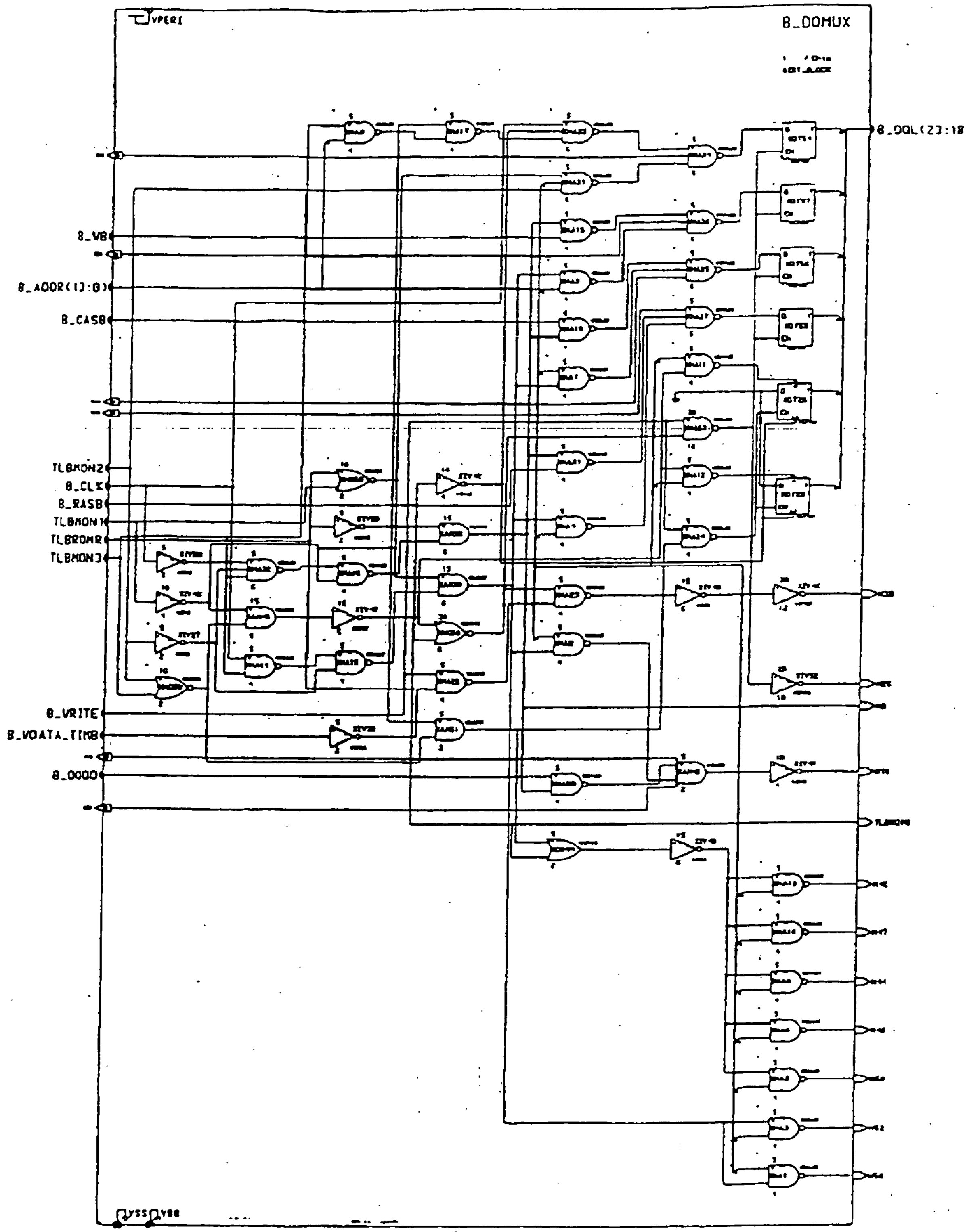

B_DOMUX
B_DOL(23:18)
B_WB
B_ADDR(13:0)
B_CASB
TLBMON2
B_CLK
B_RASB
TLBMON1
TLBRONR
TLBMON3
B_WRITE
B_WDATA_TIMB
B_0000
VPERI
VSS
VBB

FIG. 41

TI 23011

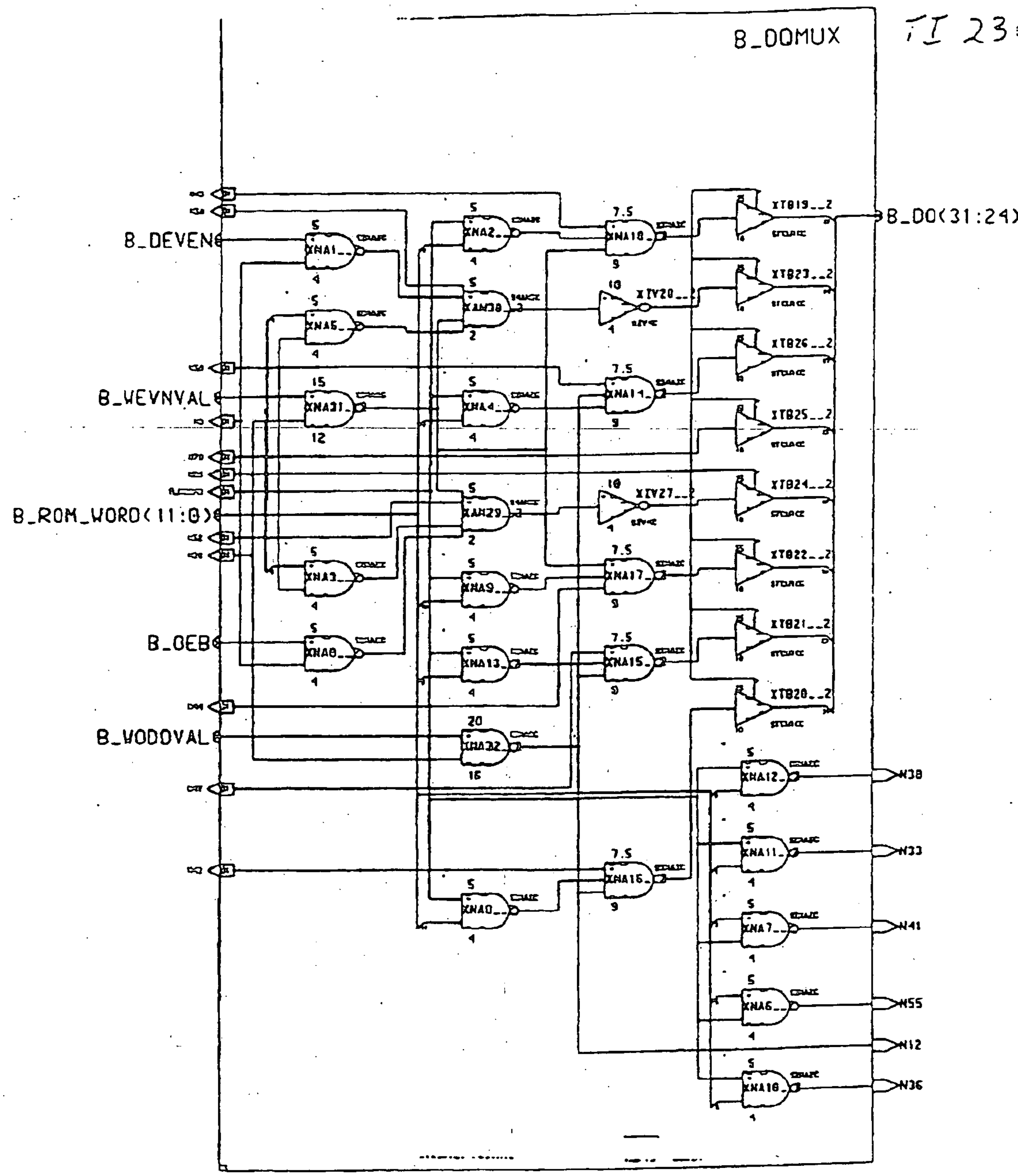


FIG. 42

TI 23 91

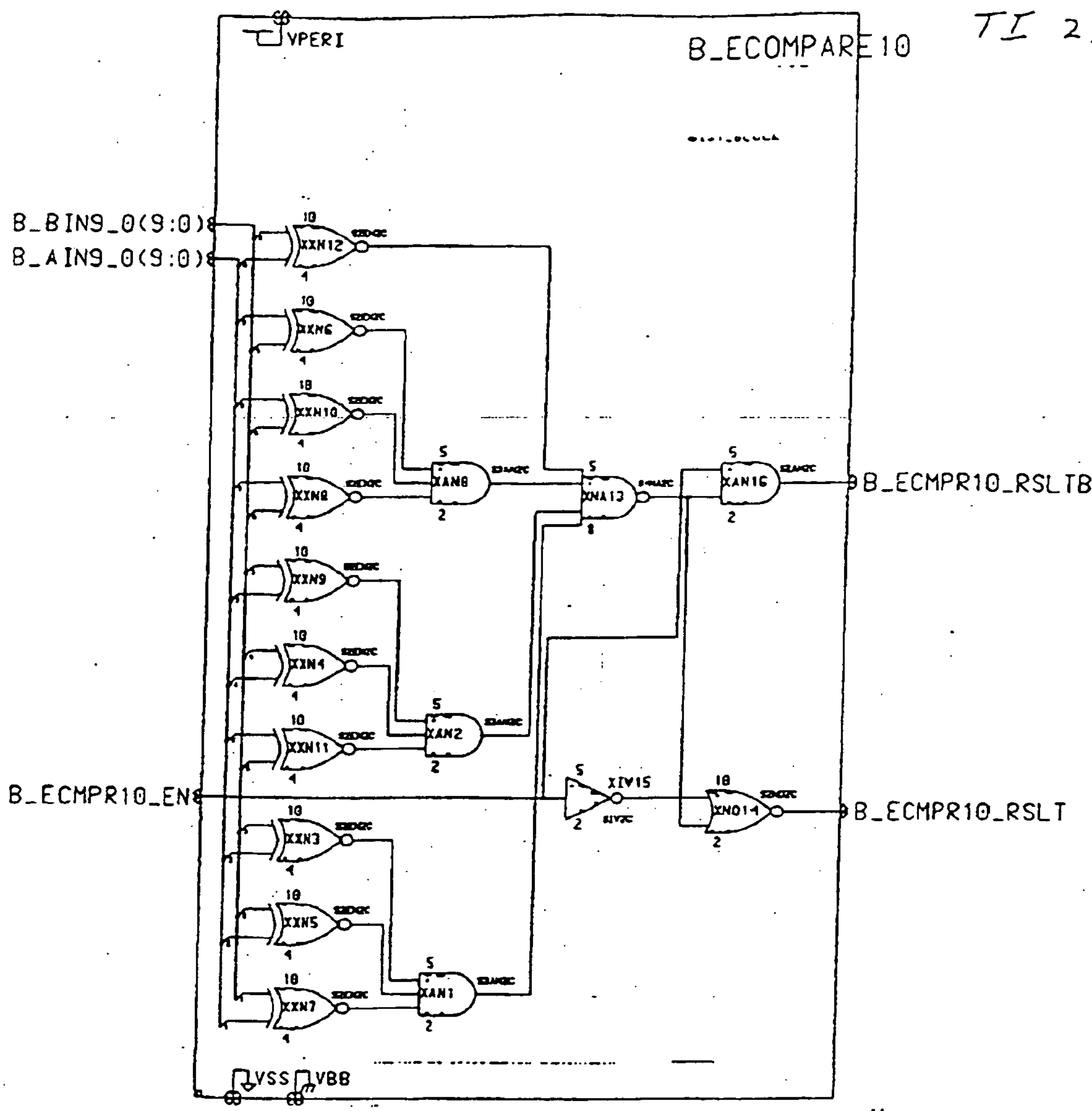


FIG. 43

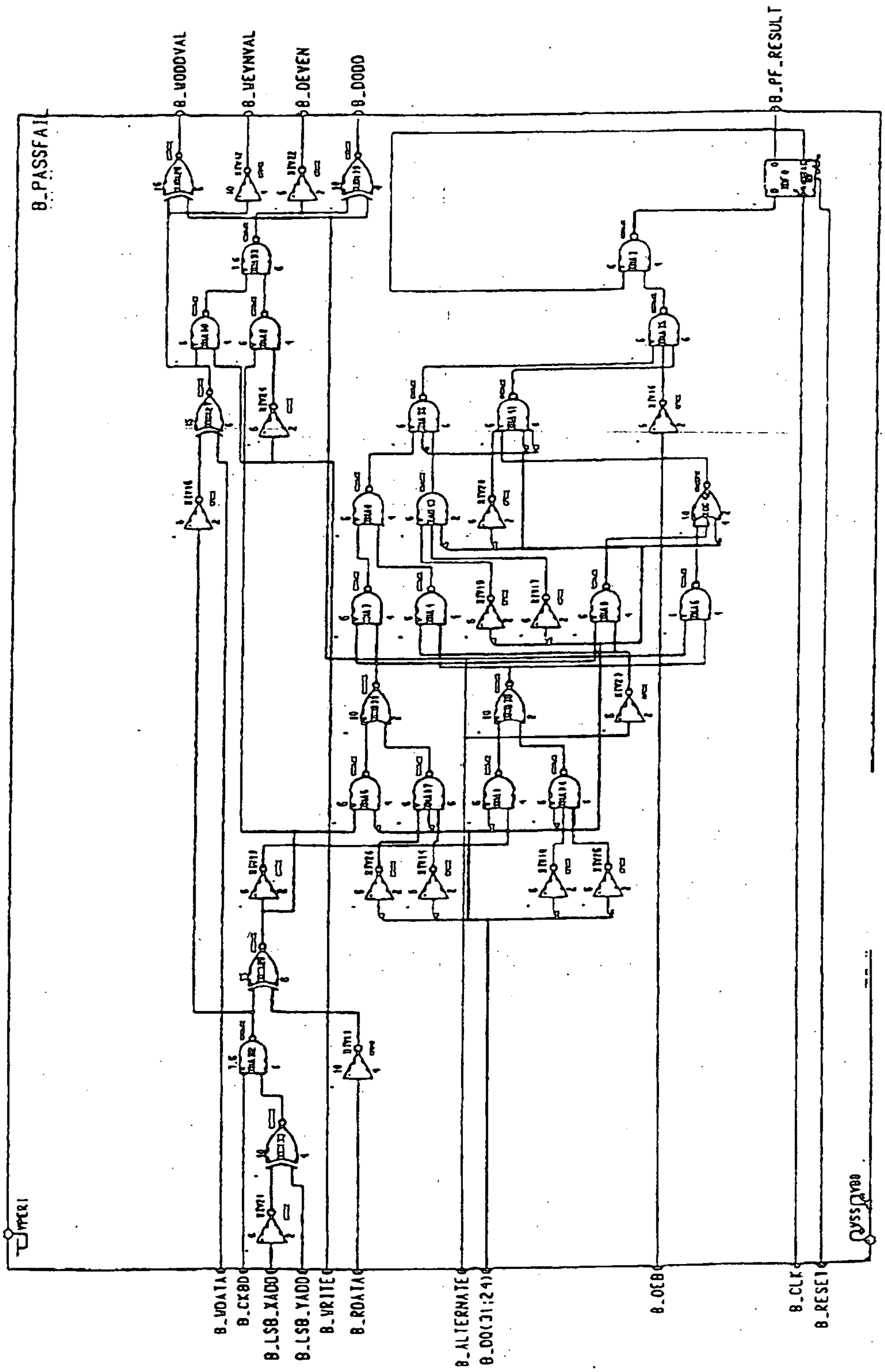

B_PASSFAIL
B_WODDVAL
B_WEVNVAL
B_DEVEN
B_DODD
B_PF_RESULT
B_WDATA
B_CKBD
B_LSB_XADD
B_LSB_YADD
B_WRITE
B_RDATA
B_ALTERNATE
B_DQ(31:24)
B_OEB
B_CLK
B_RESET
VSS
VDD

FIG. 44

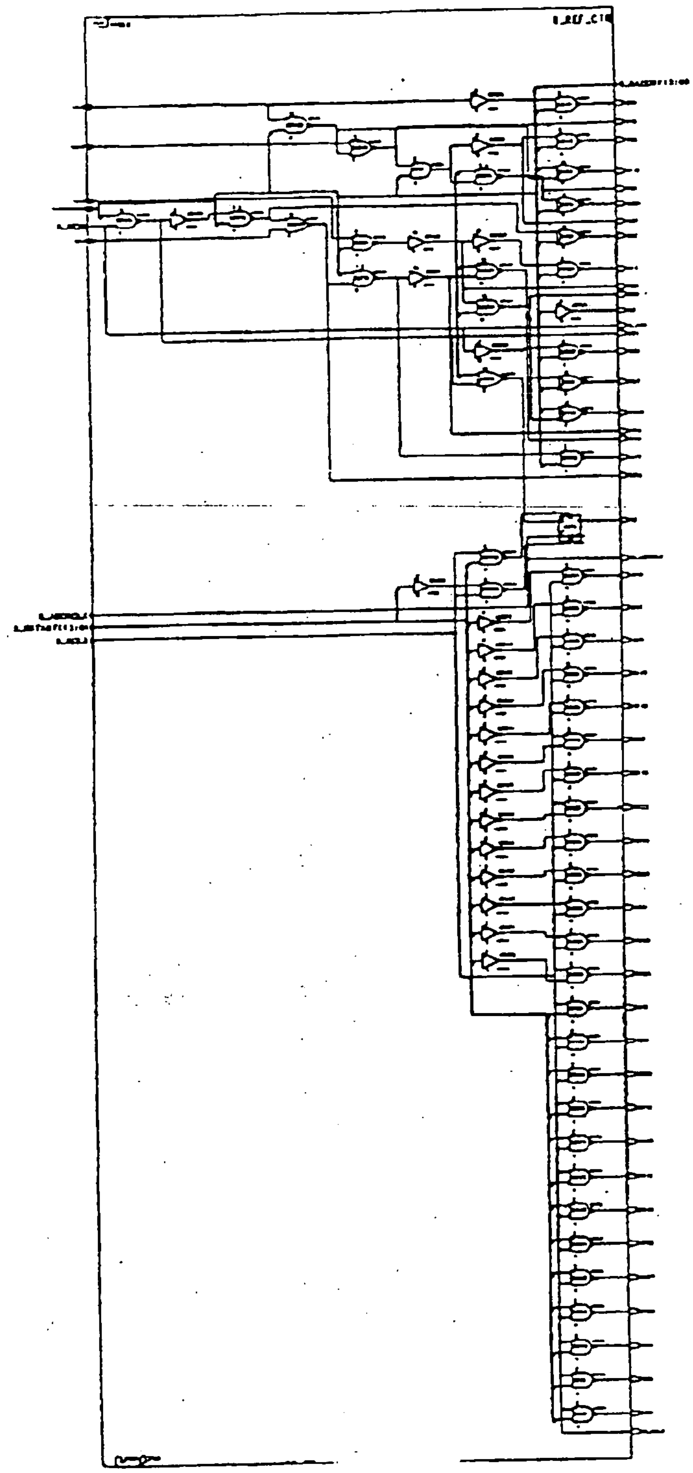


FIG. 45

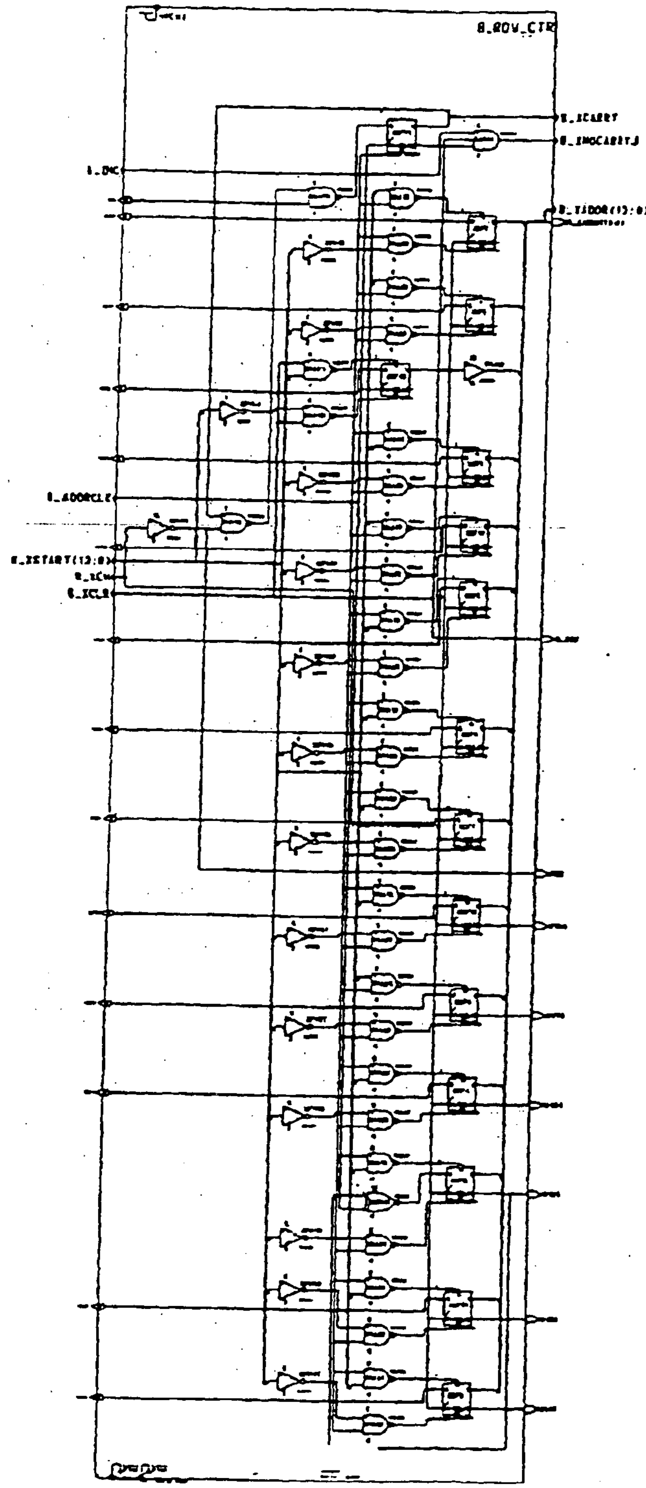


FIG. 46

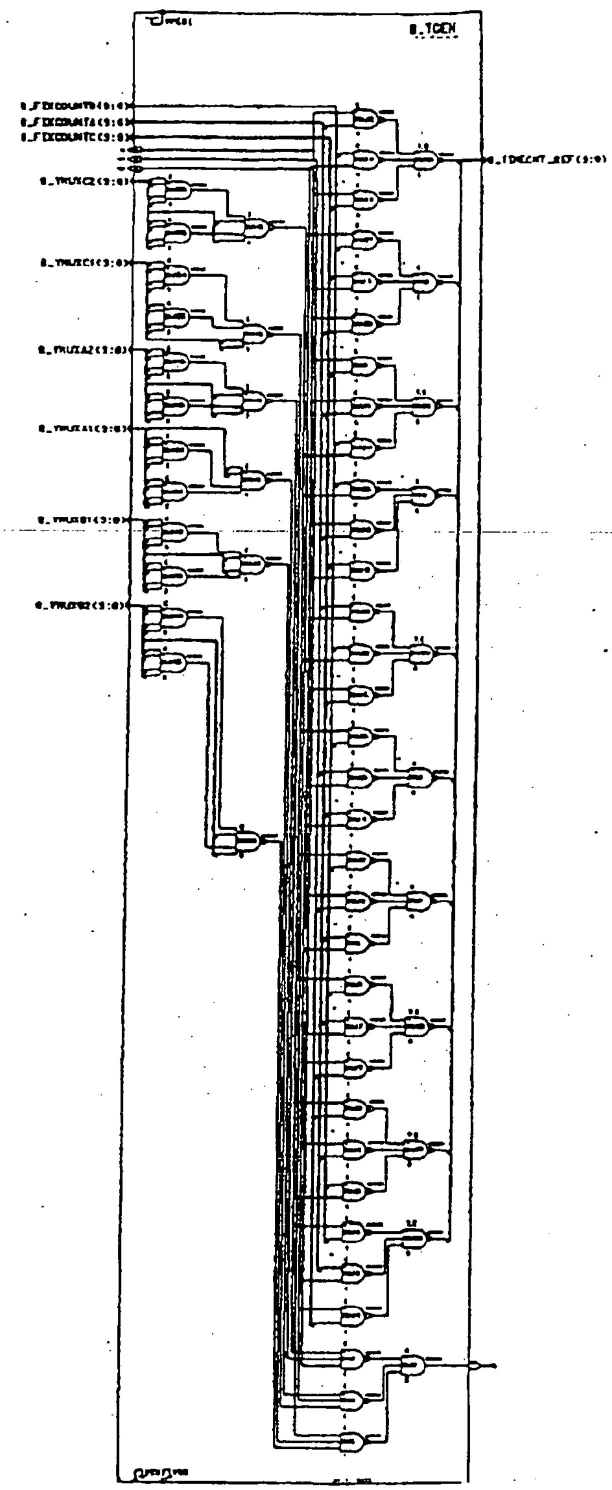


FIG. 47

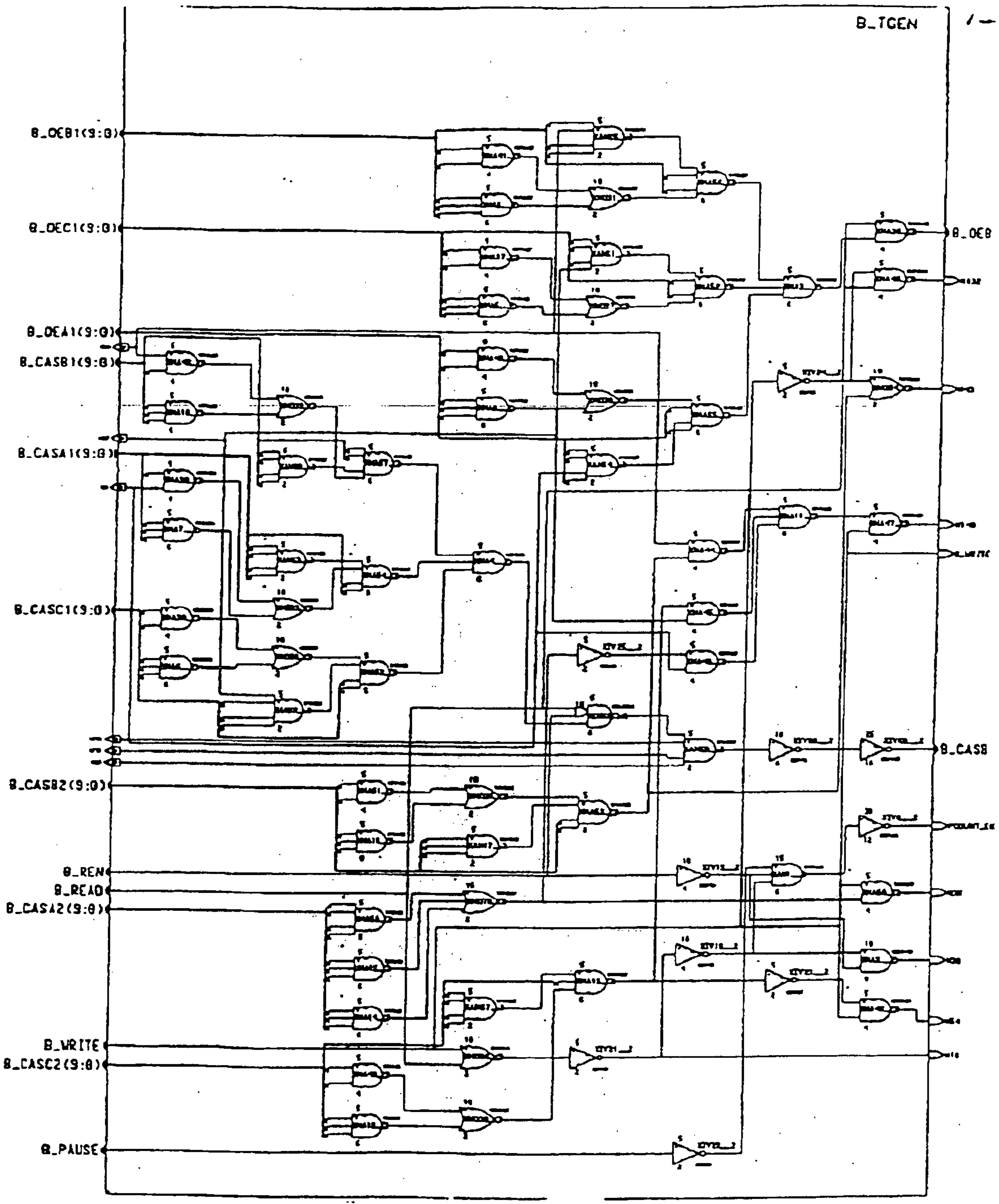

B_TGEN
B_OEB1(9:0)
B_OEC1(9:0)
B_OEA1(9:0)
B_CASB1(9:0)
B_CASA1(9:0)
B_CASC1(9:0)
B_CASB2(9:0)
B_REN
B_READ
B_CASA2(9:0)
B_WRITE
B_CASC2(9:0)
B_PAUSE
B_OEB
B_CASB

FIG. 48

TI 2

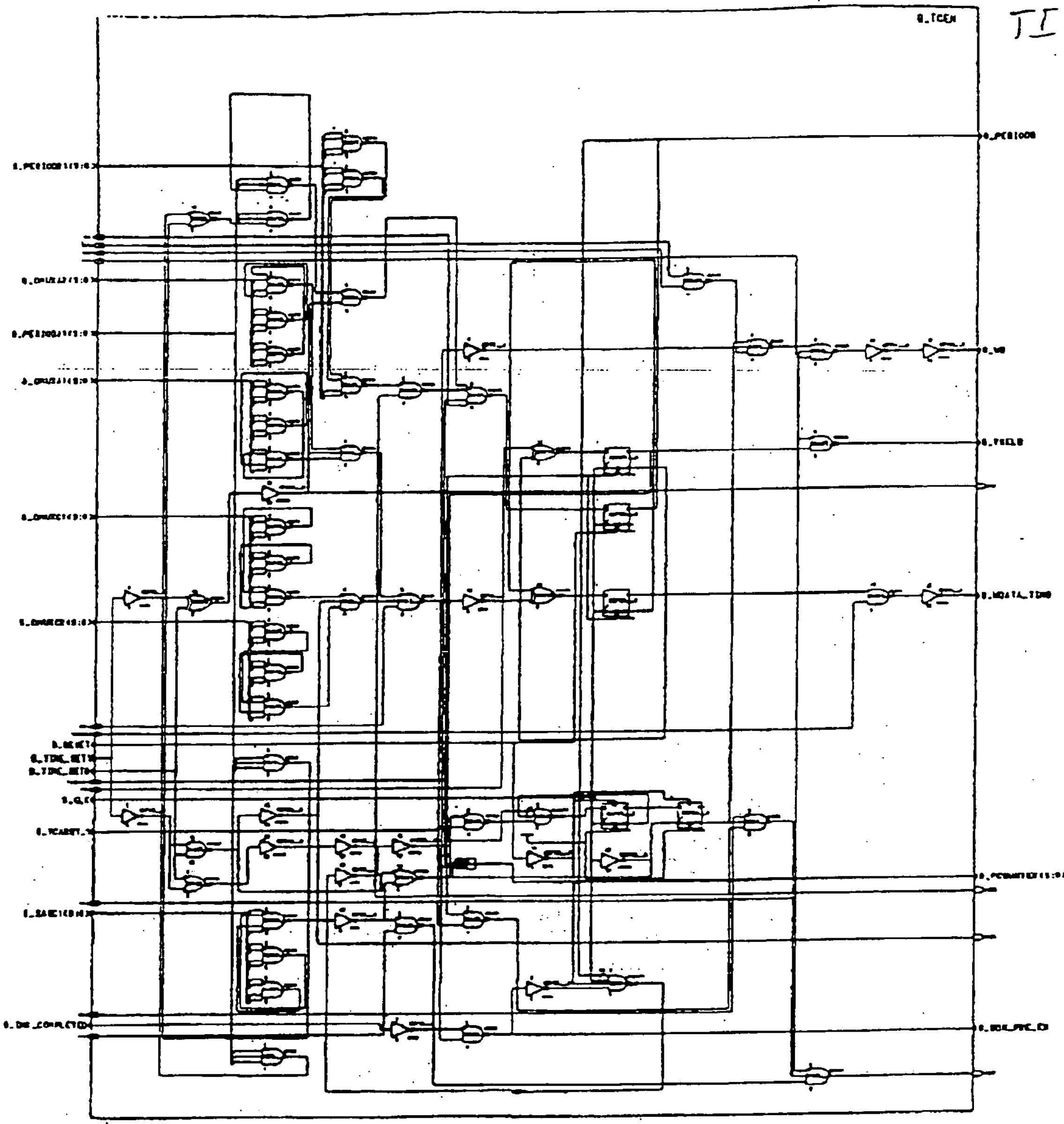


FIG. 49

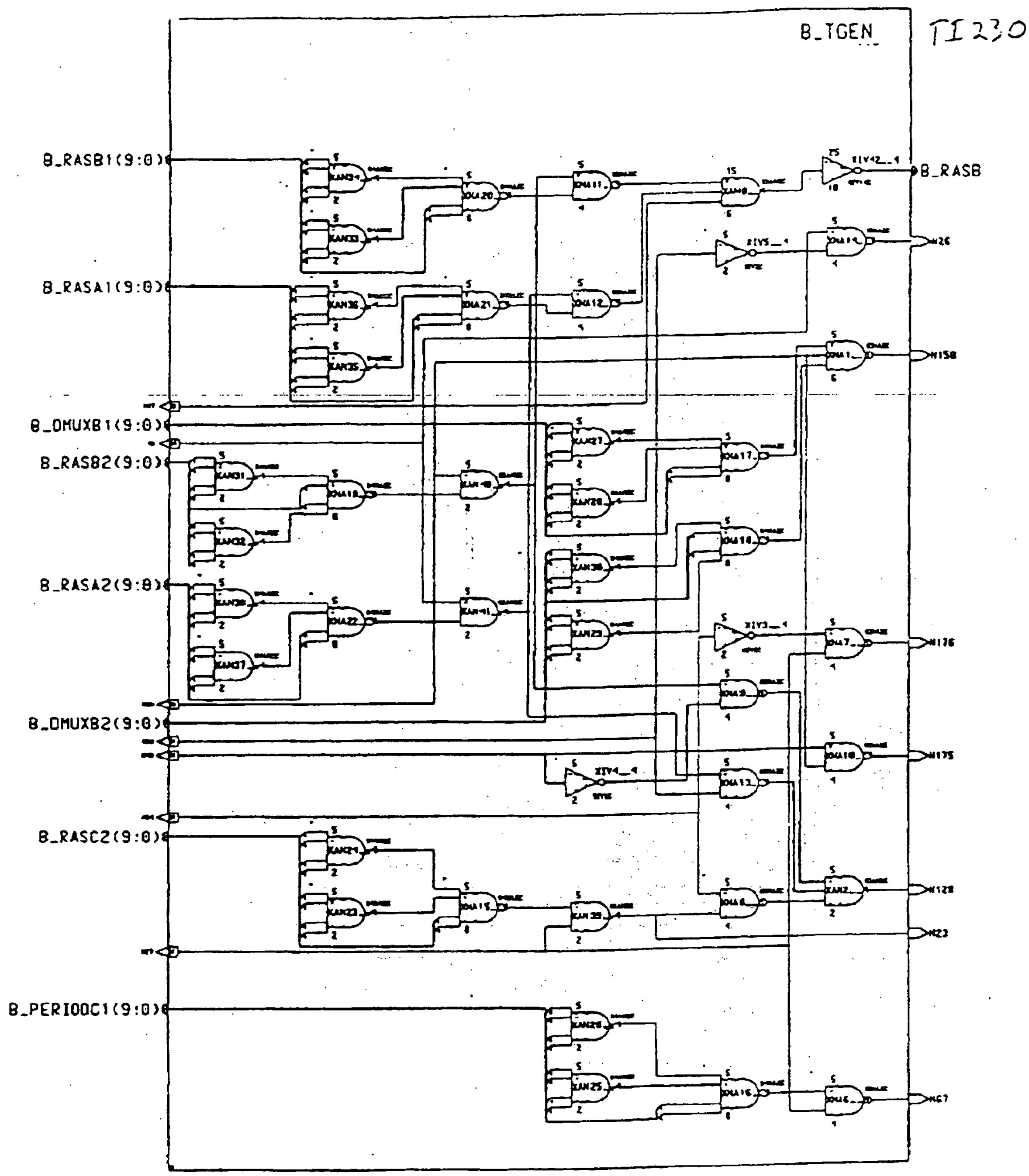

B_TGEN
TI 230
B_RASB1(9:0)
B_RASA1(9:0)
B_DMUXB1(9:0)
B_RASB2(9:0)
B_RASA2(9:0)
B_DMUXB2(9:0)
B_RASC2(9:0)
B_PERIODC1(9:0)
B_RASB

FIG. 50

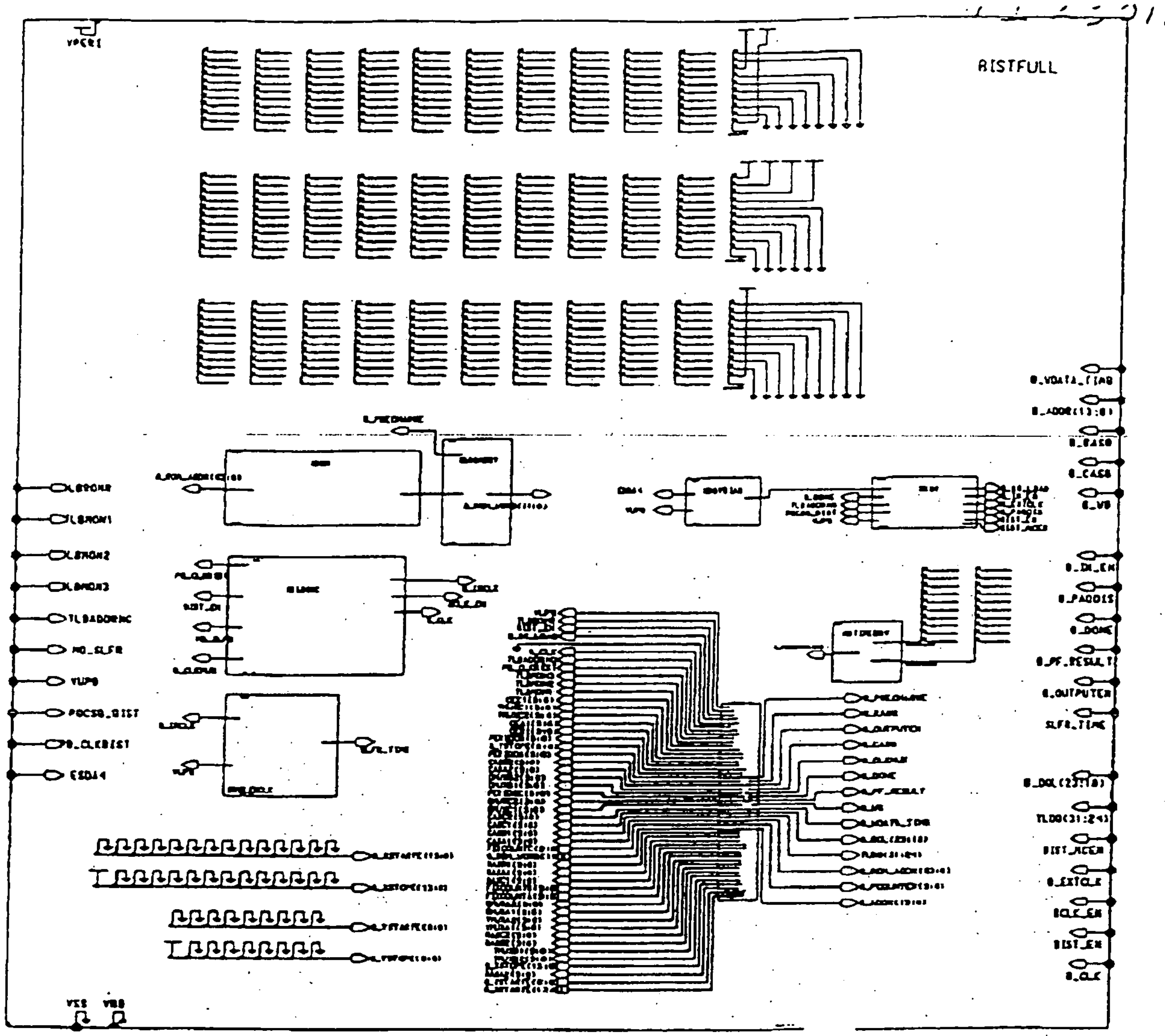


FIG. 51

TI 230

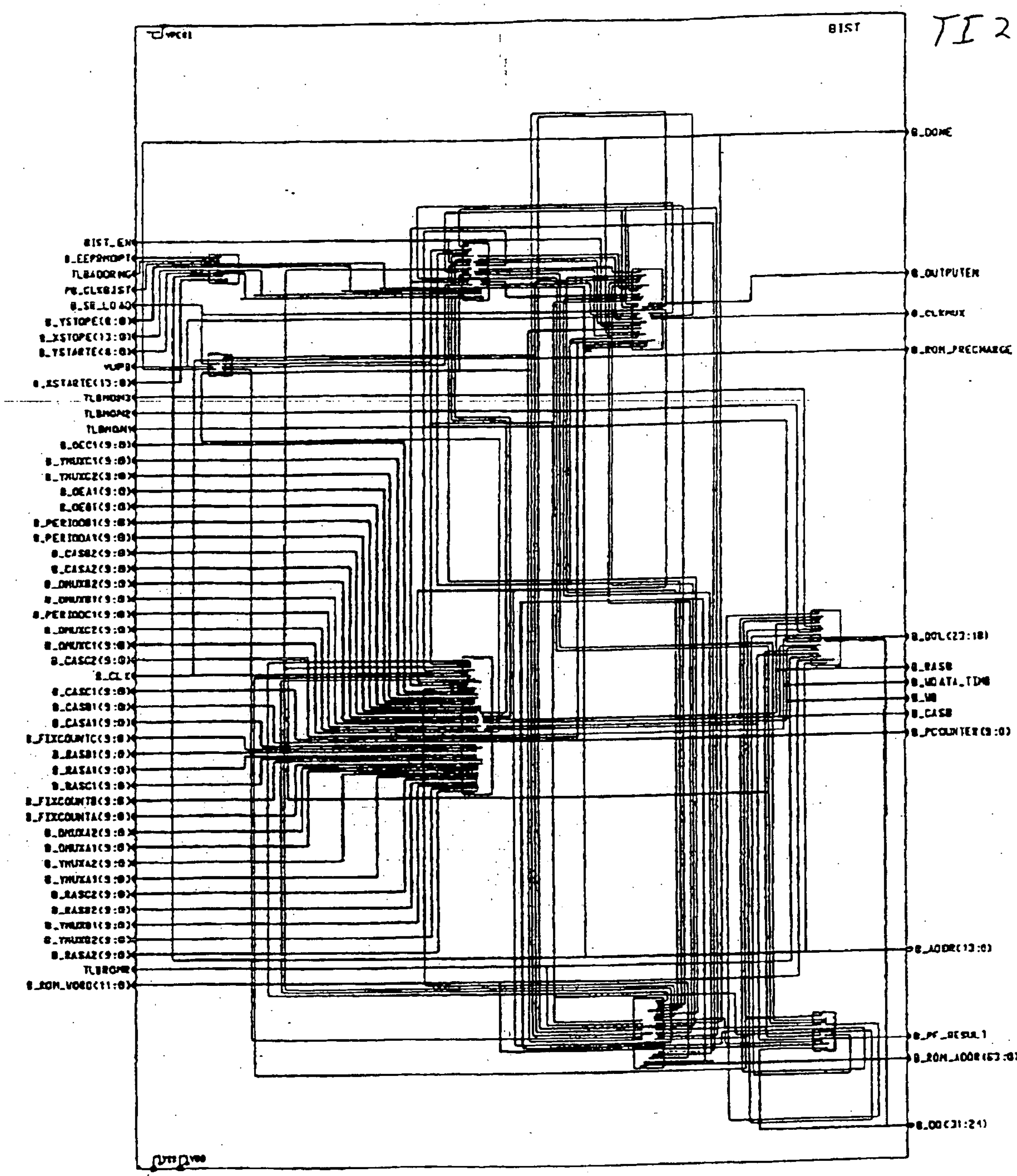
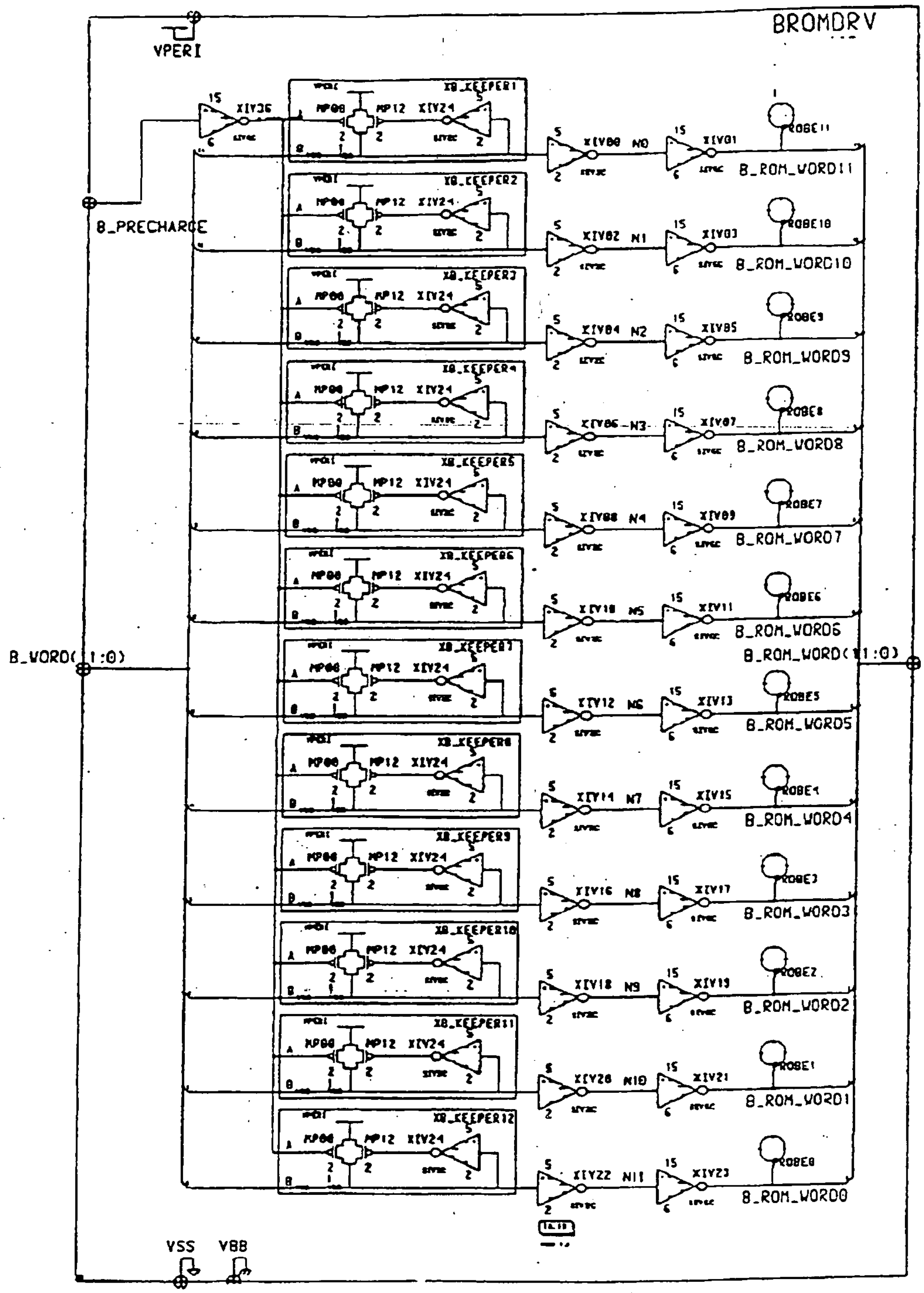

BIST
BIST_EN
B_DONE
B_OUTPUTEN
B_CLKMUX
B_ROM_PRECHARGE
B_RASB
B_WR
B_CASB
B_ADDR(13:0)
B_DO(31:24)

FIG. 52


BROMDRV
VPERI
B_PRECHARGE
B_WORD(11:0)
B_ROM_WORD(11:0)
B_ROM_WORD11
B_ROM_WORD10
B_ROM_WORD9
B_ROM_WORD8
B_ROM_WORD7
B_ROM_WORD6
B_ROM_WORD5
B_ROM_WORD4
B_ROM_WORD3
B_ROM_WORD2
B_ROM_WORD1
B_ROM_WORD0
VSS
VBB

TI 23

FIG. 53

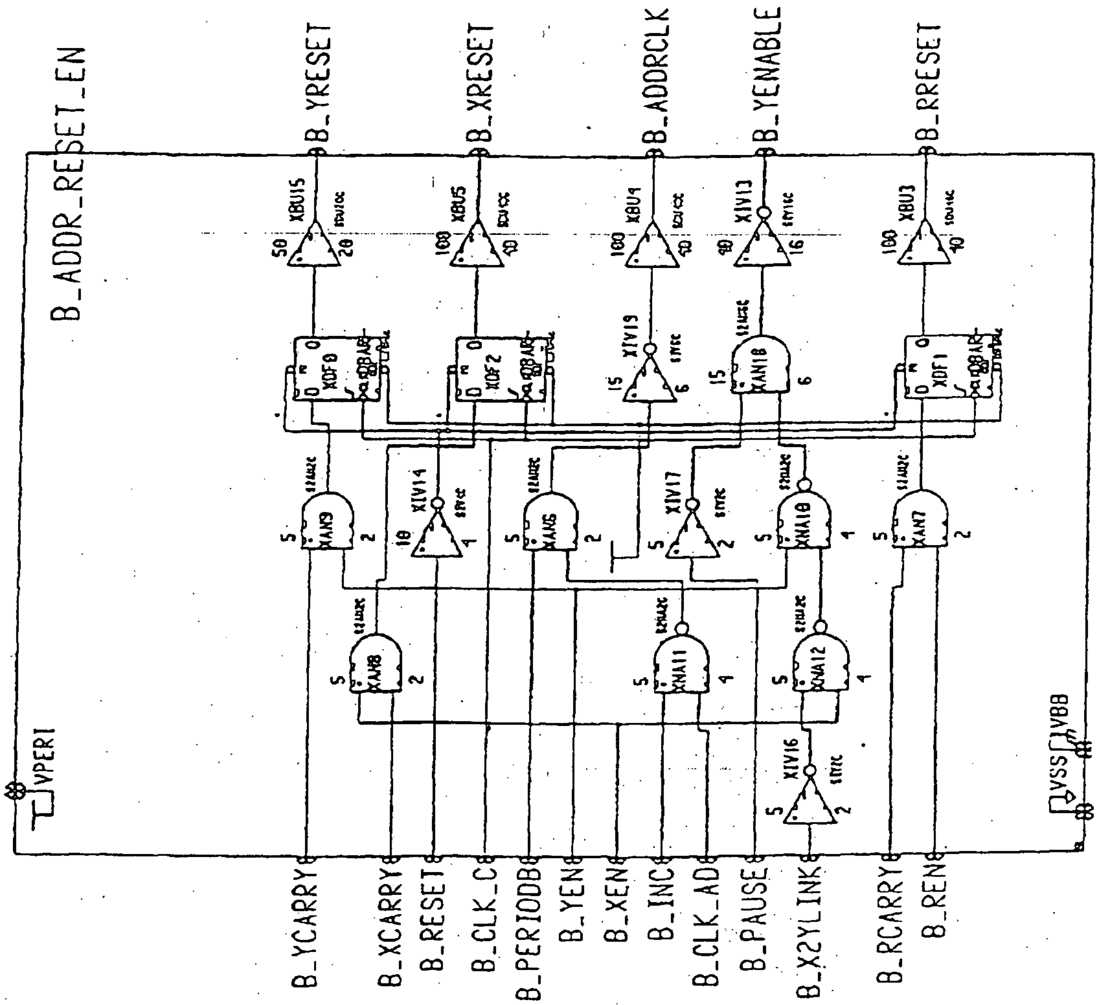

B_ADDR_RESET_EN
B_YRESET
B_XRESET
B_ADDRCLK
B_YENABLE
B_RRESET
VPERI
VSS
VBB
B_YCARRY
B_XCARRY
B_RESET
B_CLK_C
B_PERIODB
B_YEN
B_XEN
B_INC
B_CLK_AD
B_PAUSE
B_X2YLINK
B_RCARRY
B_REN

FIG. 54

TI 23011

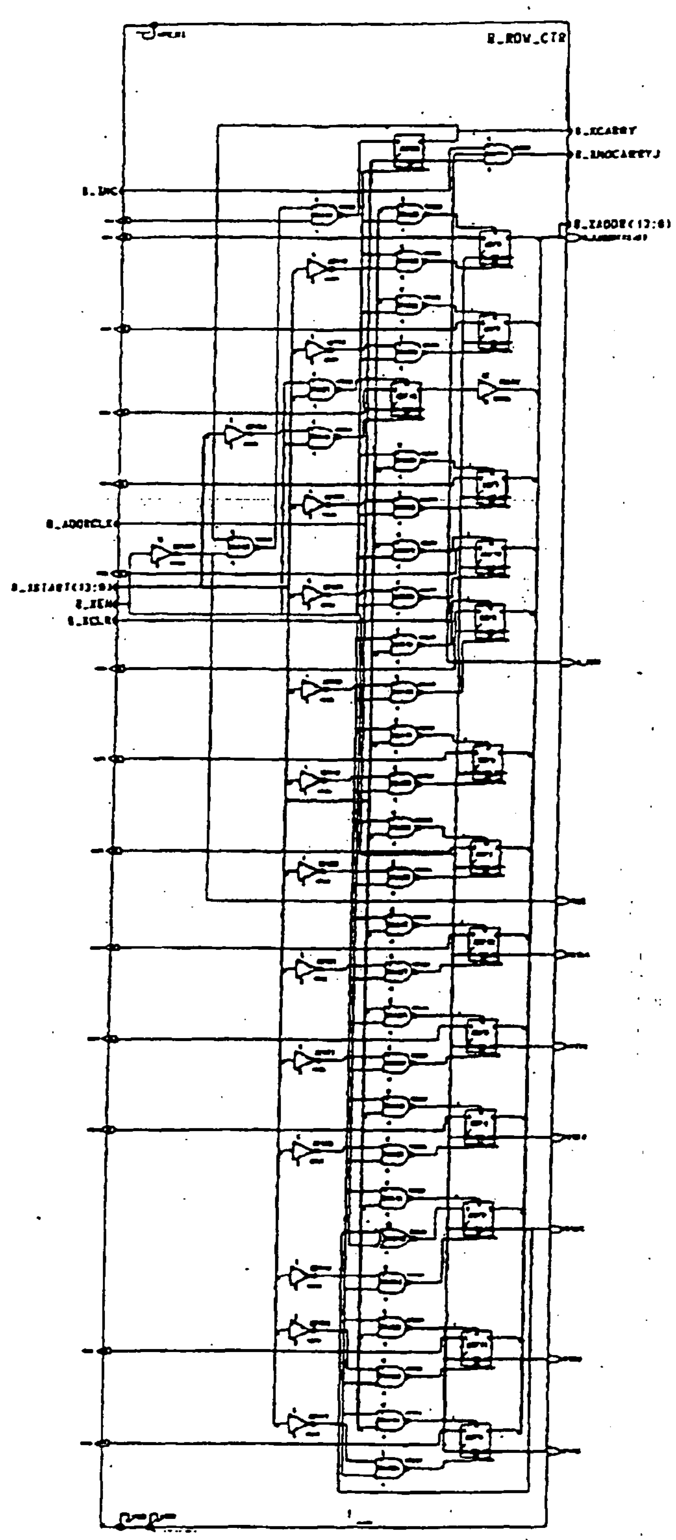
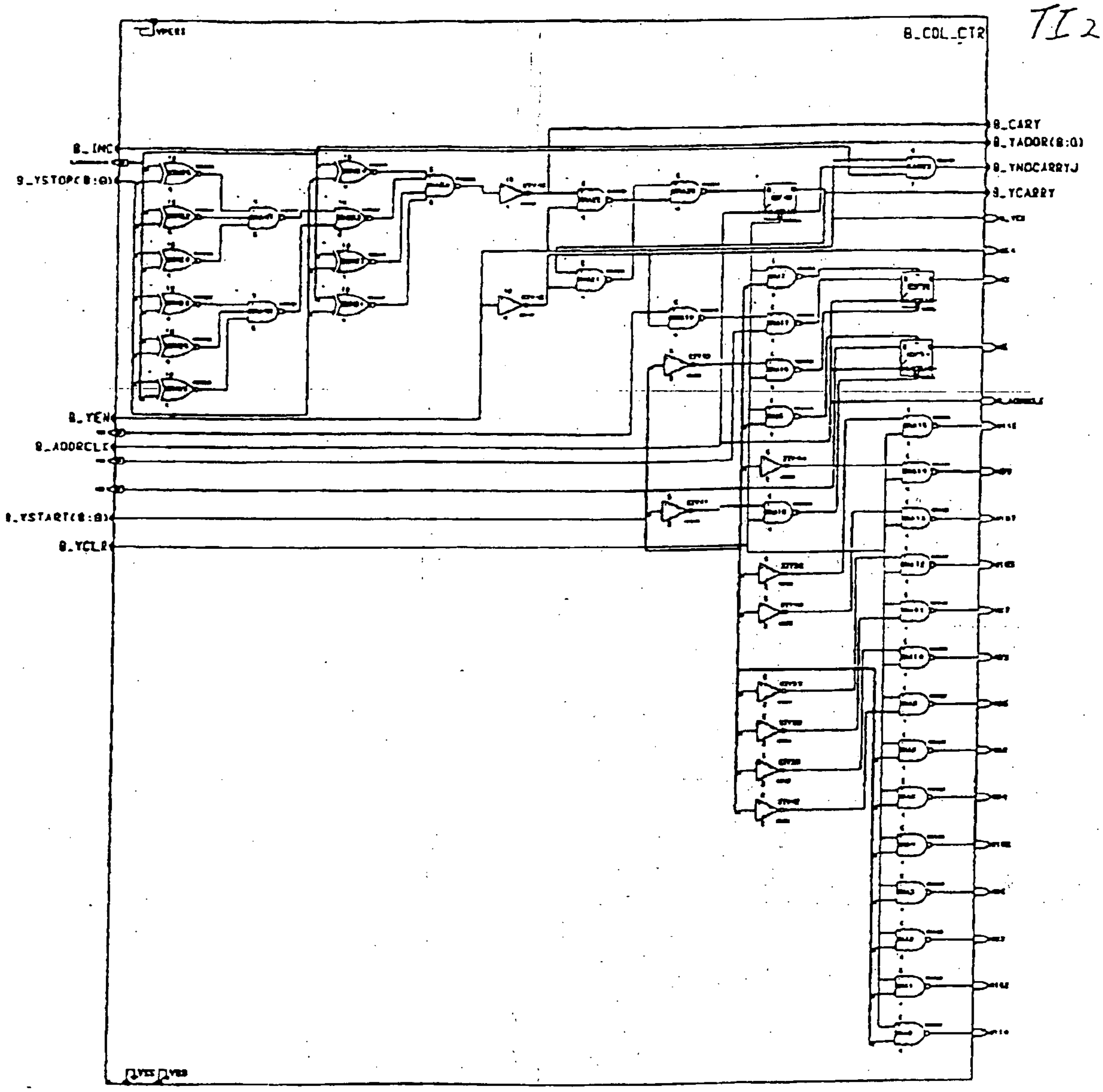


FIG. 55


B_COL_CTR
B_INC
B_YSTOP(8:0)
B_YEN
B_ADDRCLK
B_YSTART(8:0)
B_YCLR
B_CARY
B_YADDR(8:0)
B_YNOCARRYJ
B_YCARRY

TI 23c

FIG. 56

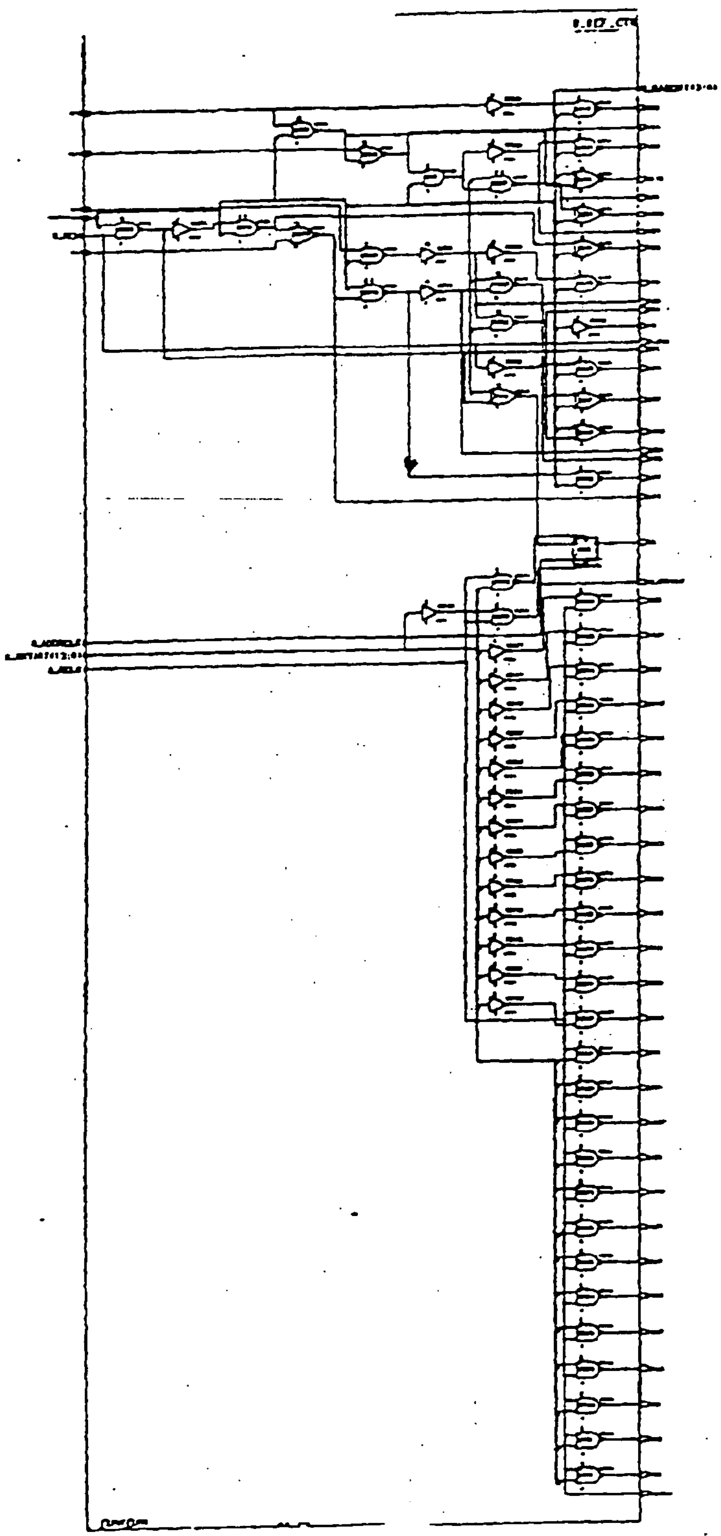


FIG. 57

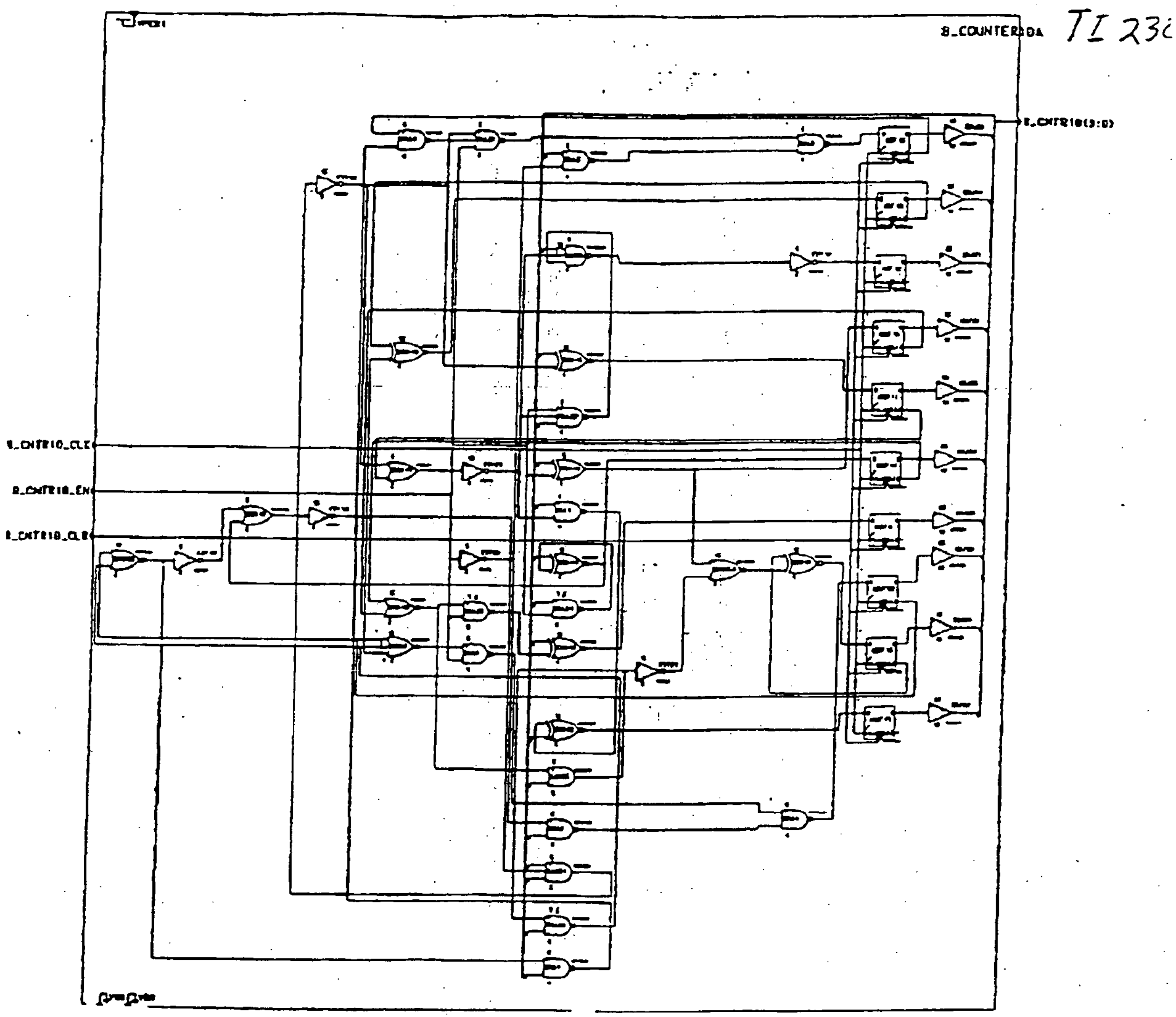


FIG. 58

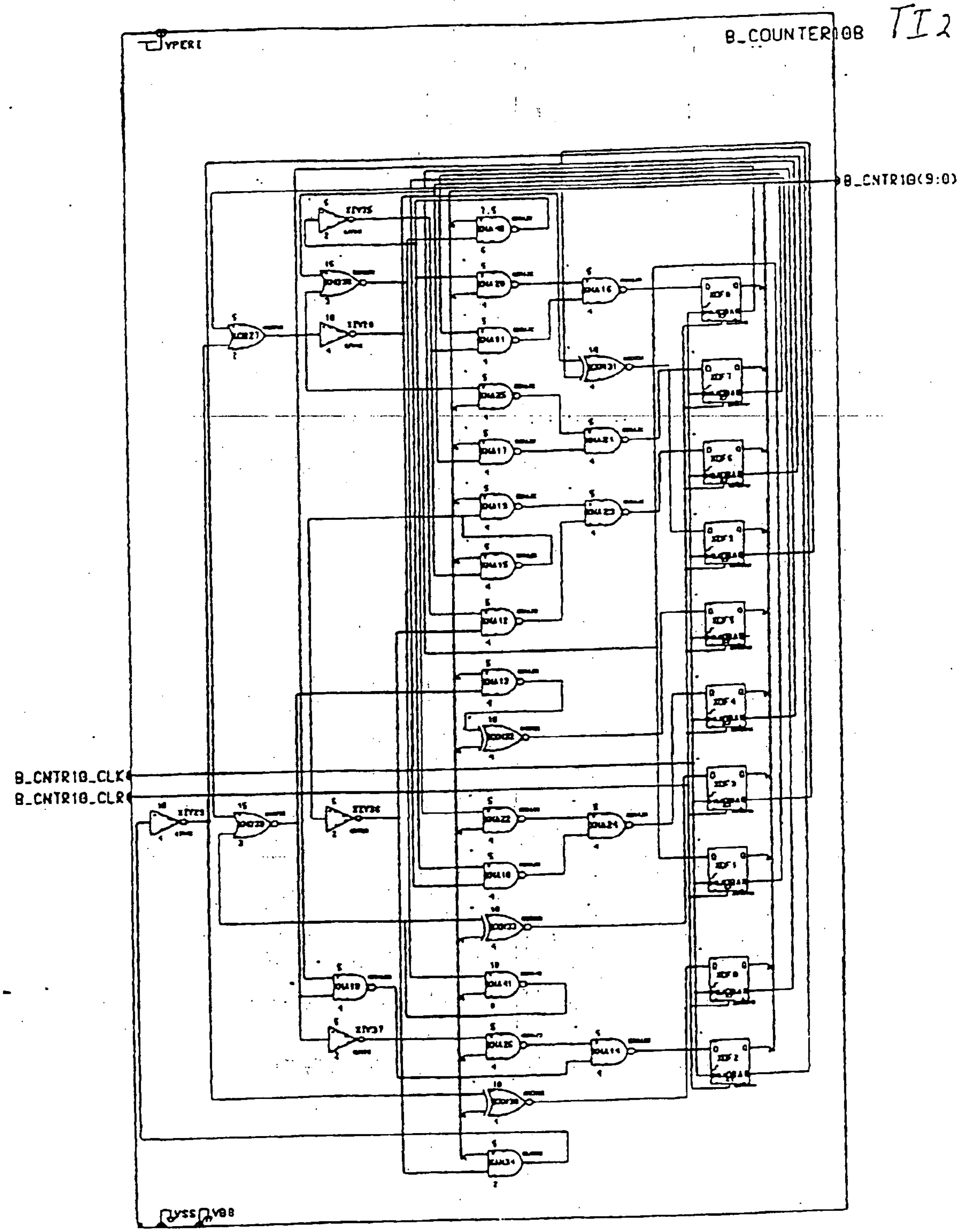

TI230
B_COUNTER10B
B_CNTR10(9:0)
B_CNTR10_CLK
B_CNTR10_CLR
VPERI
VSS
VBB

FIG. 59

Appendix: Figures and Diagrams

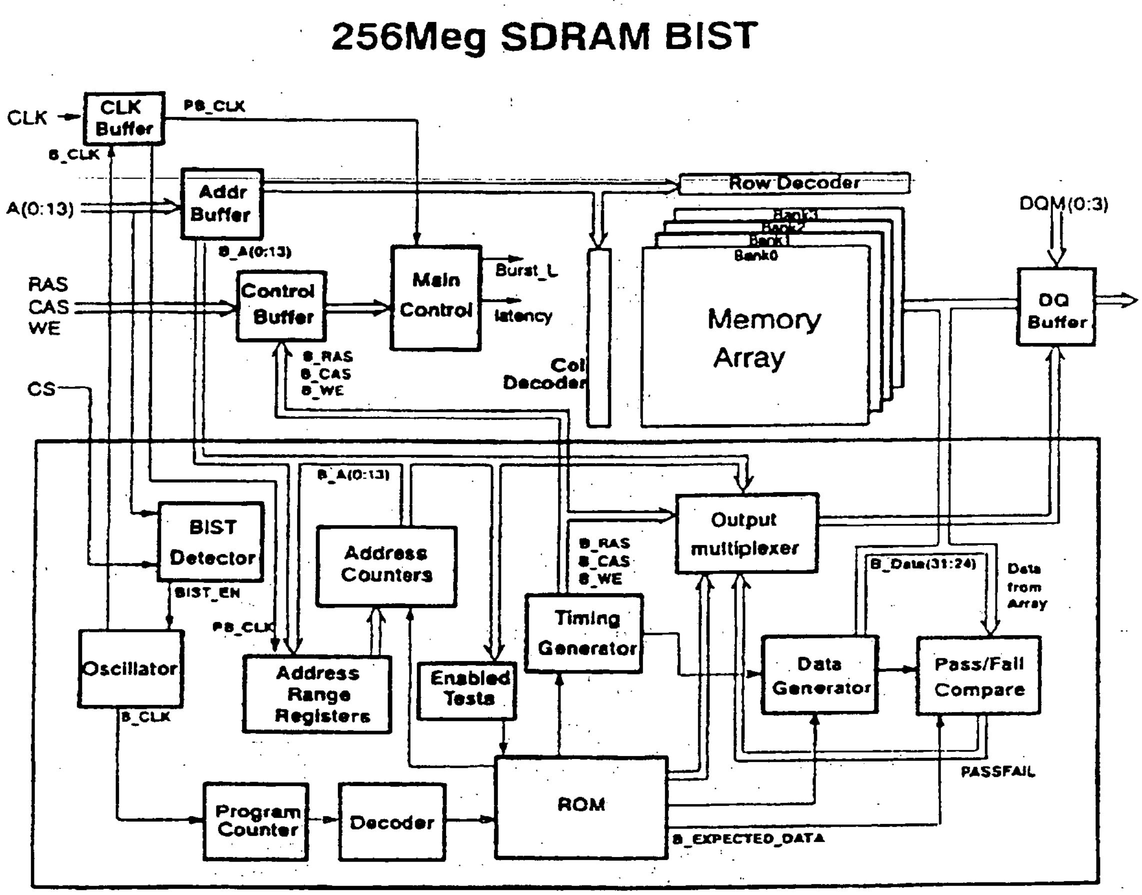


Fig 60

BIST block and Interface

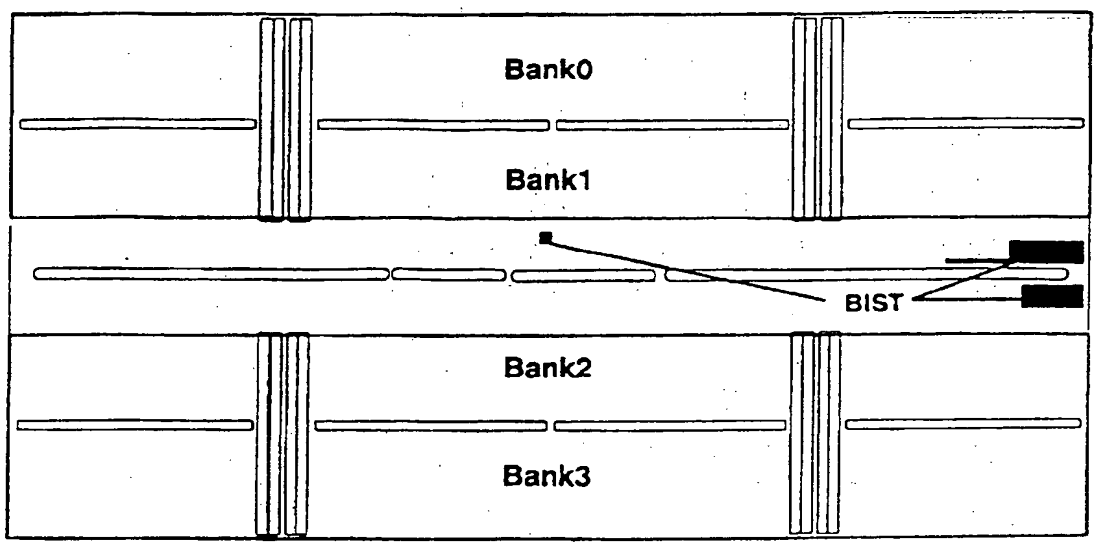


Fig 61

Algorithms	Timing Set	Addr Access	Array Access	Data Pattern	Misc option
Gross	Short	X, inc, dec	R, W	0, 1	CKBD, ALT
Pause	Short	X, Inc	R, W	0, 1	CKBD, Pause
Xmarch	Long	X, inc	RW	0, 1	
Ymarch	Short	Y, inc	RW	0, 1	
SDIST	Short	X,Y, inc	R, W	0, 1	Refresh
LDIST	Long	X,Y, inc	R, W	0, 1	Refresh
PDIST	Page	Y, inc	R, W	0, 1	Refresh
Burn In	Long	X,Y, inc	W	0, 1	CKBD
Eng 1	Short	Y, inc, 1 row	W	1	
Eng2	Short	X, inc, 1 Col	R	0	

Short – 300ns period
Long – 10,000ns period

R – Read
W – Write

inc – Increment
dec – Decrement

Fig 62

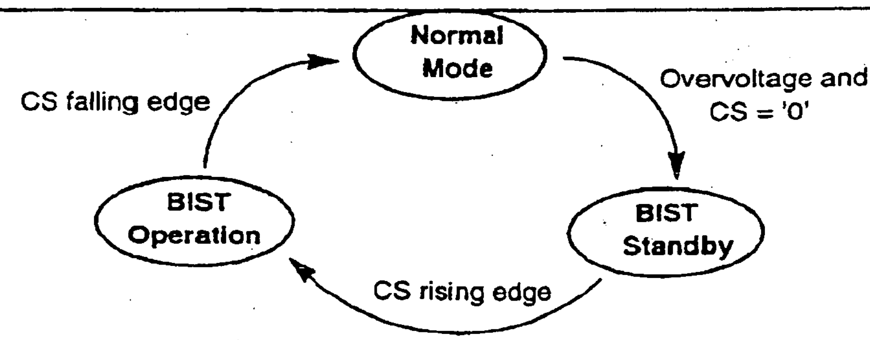


Fig 63

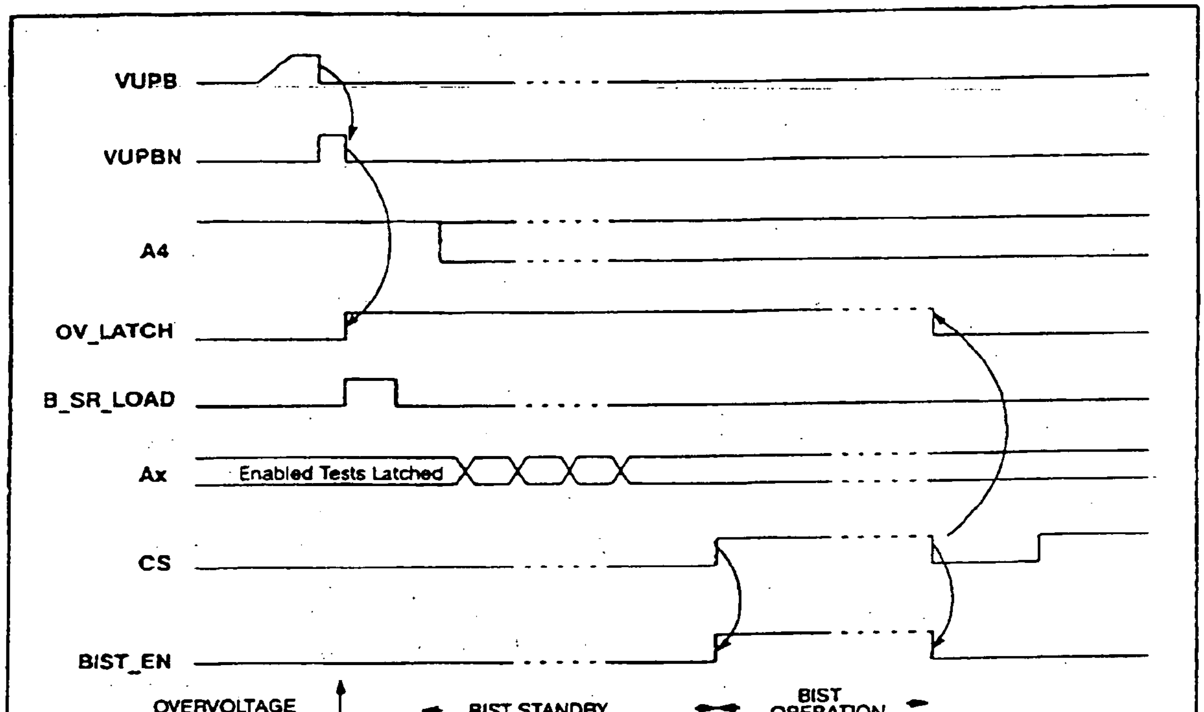

VUPB
VUPBN
A4
OV_LATCH
B_SR_LOAD
Ax
Enabled Tests Latched
CS
BIST_EN
OVERVOLTAGE DETECTION
BIST STANDBY
BIST OPERATION

Fig 64

256M SDRAM

3.3V	VDD			VSS	0.0V
No connect	DQ0	2	87	DQ15	No connect
3.3V	VDDQ	3	86	VSSQ	0.0V
No connect	DQ1	4	85	DQ14	No connect
No connect	DQ2	5	84	DQ13	No connect
0.0V	VSSQ	6	83	VDDQ	3.3V
No connect	DQ3	7	82	DQ12	No connect
No connect	DQ4	8	81	DQ11	No connect
3.3V	VDDQ	9	80	VSSQ	0.0V
No connect	DQ5	10	79	DQ10	No connect
No connect	DQ6	11	78	DQ9	No connect
0.0V	VSSQ	12	77	VDDQ	3.3V
No connect	DQ7	13	76	DQ8	No connect
0.0V	DQM0	14	75	DQM1	0.0V
3.3V	VDD	15	74	VSS	0.0V
No connect	NC	16	73	NC	No connect
3.3V	W	17	72	NC	No connect
3.3V	CE	18	71	Vref	1.65V
3.3V	RE	19	70	NC	No connect
0.0V / 3.3V	S	20	69	CLK	0.0V
0.0V / 3.3V	BA1	21	68	CKE	3.3V
0.0V / 3.3V	A11	22	67	BA0	0.0V / 3.3V
3.3V	VDD	23	66	VSS	0.0V
0.0V / 3.3V	A0	24	65	NC	No connect
0.0V / 3.3V	A1	25	64	A10	0.0V / 3.3V
0.0V / 3.3V	A2	26	63	A9	0.0V / 3.3V
0.0V / 3.3V	A3	27	62	A8	0.0V / 3.3V
0.0V / 5.4V	A4	28	61	A7	0.0V / 3.3V
0.0V / 3.3V	A5	29	60	A6	0.0V / 3.3V
3.3V	VDD	30	59	VSS	0.0V
0.0V	DQM2	31	58	DQM3	0.0V
DONE	DQ16	32	57	DQ31	No connect
0.0V	VSSQ	33	56	VDDQ	3.3V
PASSFAIL	DQ17	34	55	DQ30	No connect
No connect	DQ18	35	54	DQ29	No connect
3.3V	VDDQ	36	53	VSSQ	0.0V
No connect	DQ19	37	52	DQ28	No connect
No connect	DQ20	38	51	DQ27	No connect
0.0V	VSSQ	39	50	VDDQ	3.3V
No connect	DQ21	40	49	DQ26	No connect
No connect	DQ22	41	48	DQ25	No connect
3.3V	VDDQ	42	47	VSSQ	0.0V
No connect	DQ23	43	46	DQ24	No connect
3.3V	VDD			VSS	0.0V

Fig 65

	Normal Mode	BIST : Int Clk (B_CLKMUX = 0)	BIST : Ex Clk (B_CLKMUX = 1)
B_CLK	Low	20ns Clock	Same as Ext Clk
B_IRCLK	0.5Us Clock	Low	Low

Fig 66

256M BIST ALGO STORAGE CONFIGURATION

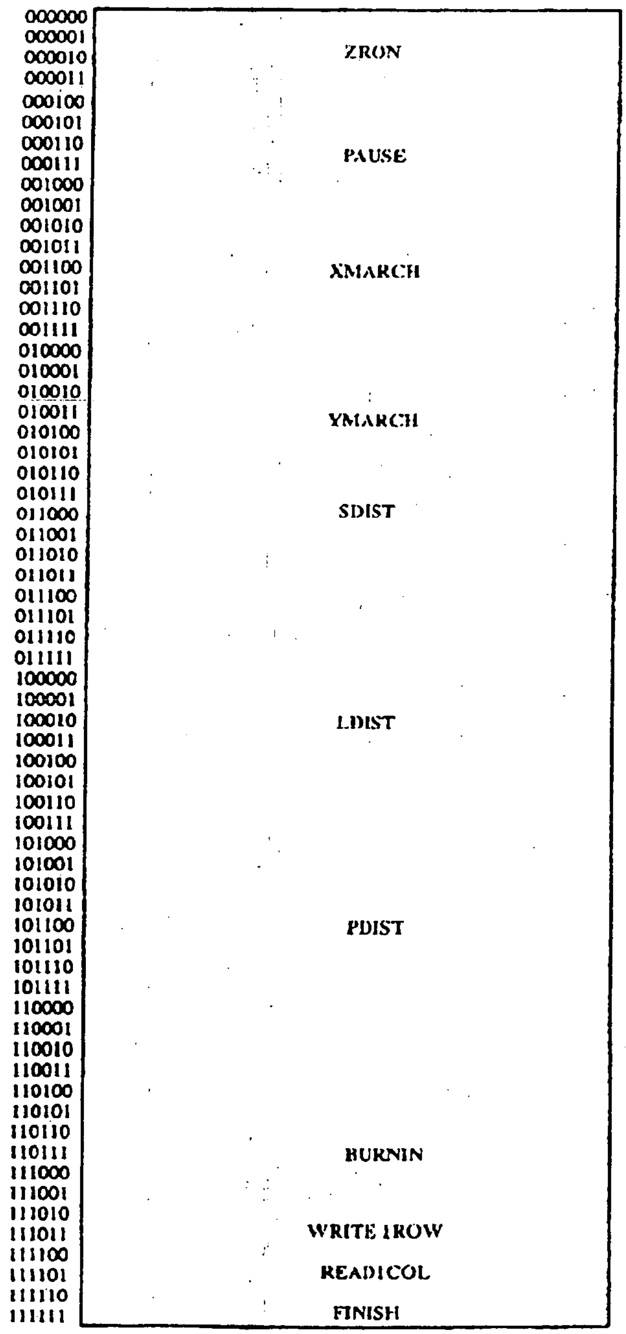


Fig 67

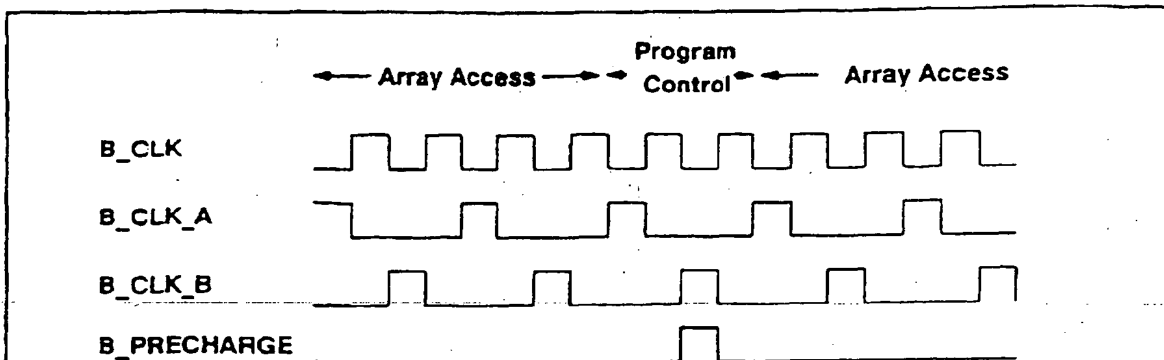


Fig 68

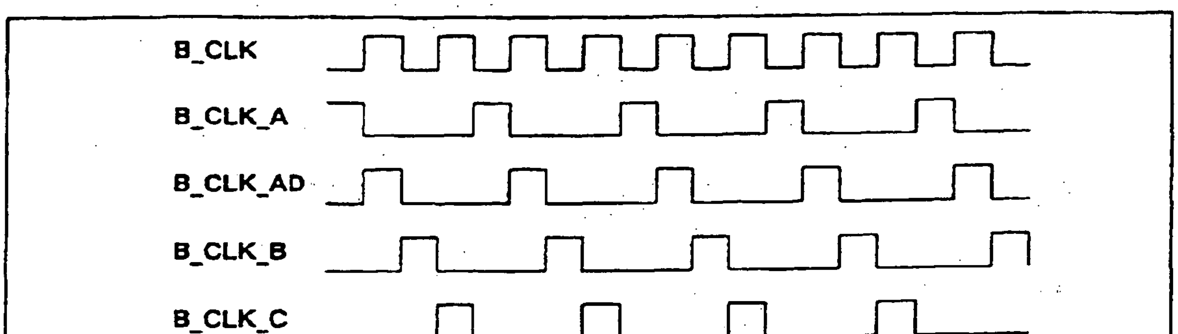


Fig 69

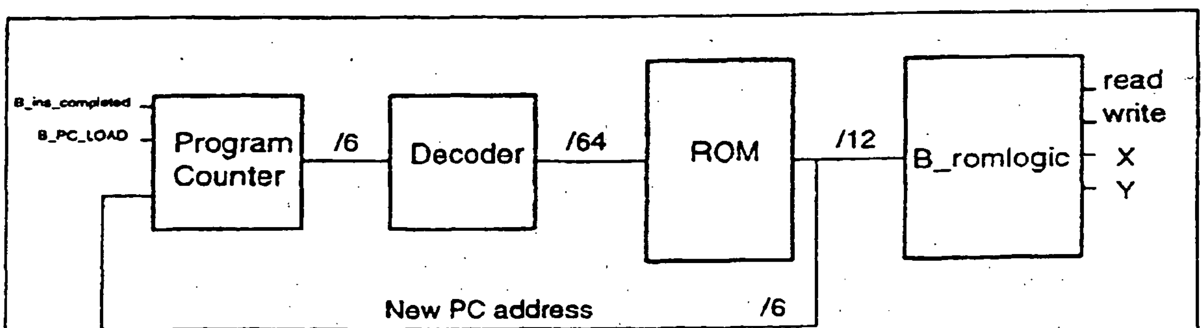
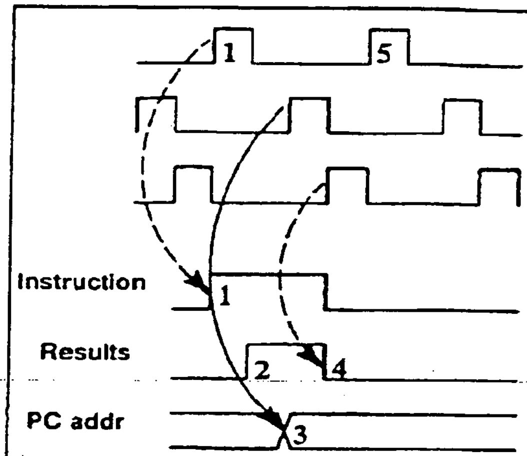
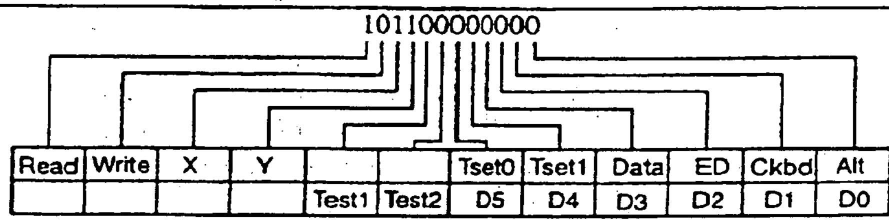


Fig 70



1. Instructions appears at CLK_A R/E
2. Results of operations appears at CLK_A F/E
3. At CLK_B R/E, decide on new PC address
4. At rising edge of CLK_C, resets all signals
5. New Instructions appears at CLK_A R/E

Fig 71



Array Access
Program control

Test1	Test2	Comments
0	0	Test if the test is enabled
0	1	Test for inverted pattern
1	0	Increment address by 1, and check for overflow
1	1	Increment counter and check if required value is reached

Tset0	Tset1	Comments
0	0	Timing set A
0	1	Timing set C
1	0	Timing set B
1	1	Timing set A, decrement

Fig 72

Timing sets	Access mode	Addressing	Array size	'0' or '1'	Data Pattern
TSETA	READ	INC	FULLA	PATTERN0	CKBD
TSETB	WRITE	DEC	ROW	PATTERN1	ALTERNATE
TSETC	RMW		COL		

Fig 73

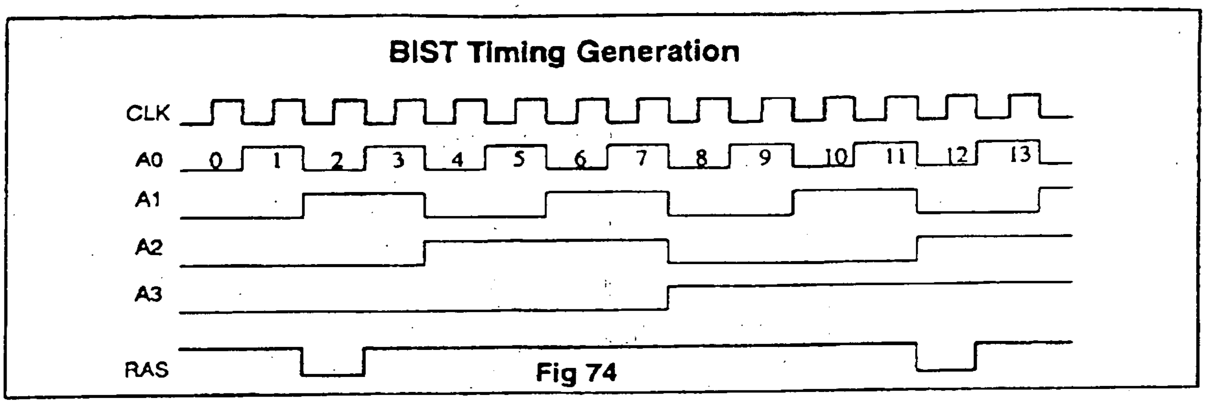
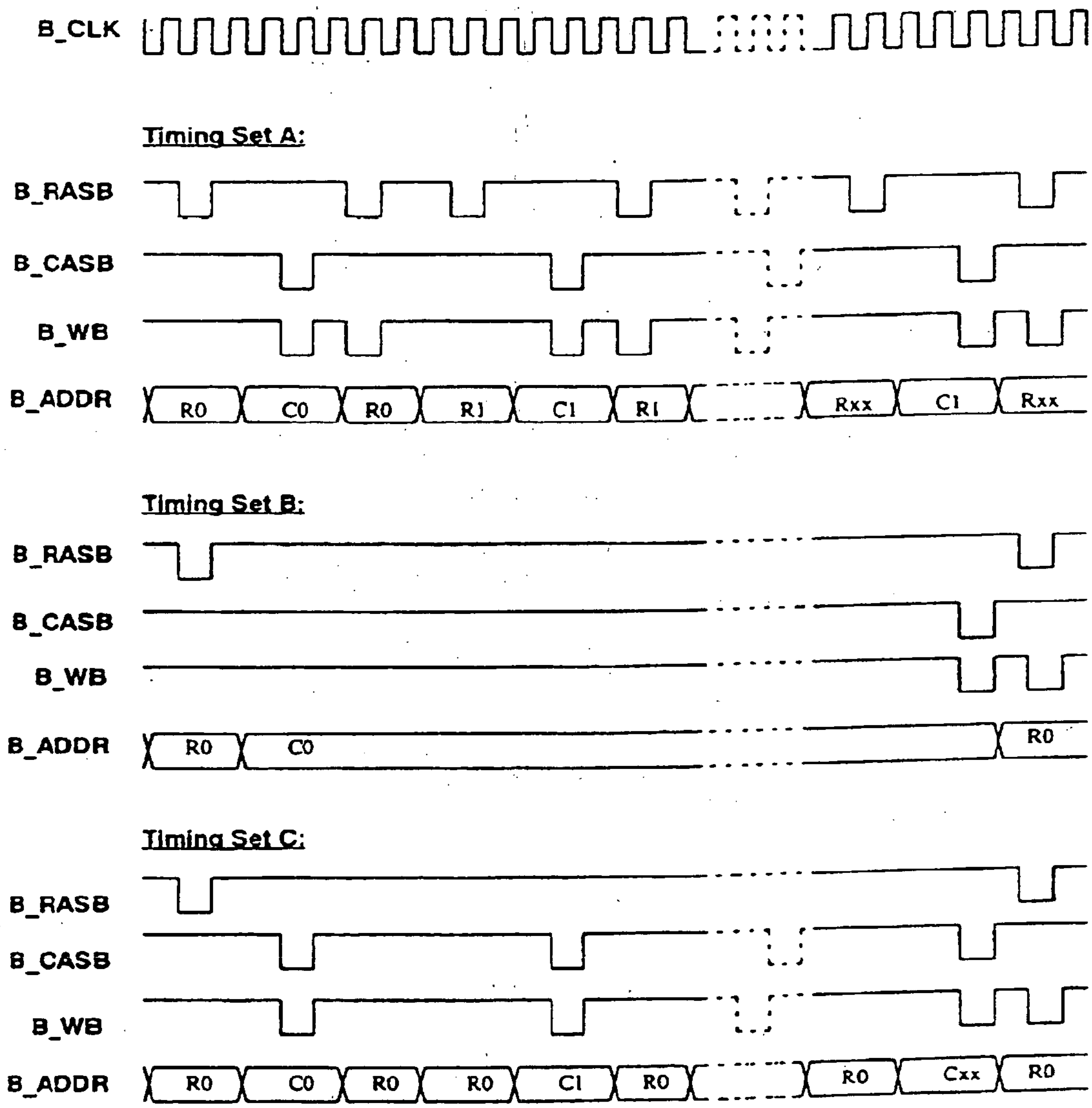

BIST Timing Generation
CLK
A0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
A1
A2
A3
RAS

Fig 74

BIST Timing Sets



Timing Set	Cycle Time	Description
A	300ns	Short timing, used in Gross,Pause,Ymarch,SDIST,Eng1,Eng2
B	10,000ns	Long Timing, used in Xmarch, LDIST, Burn In
C	300ns	Multiple columns, used in PDIST

Fig 75

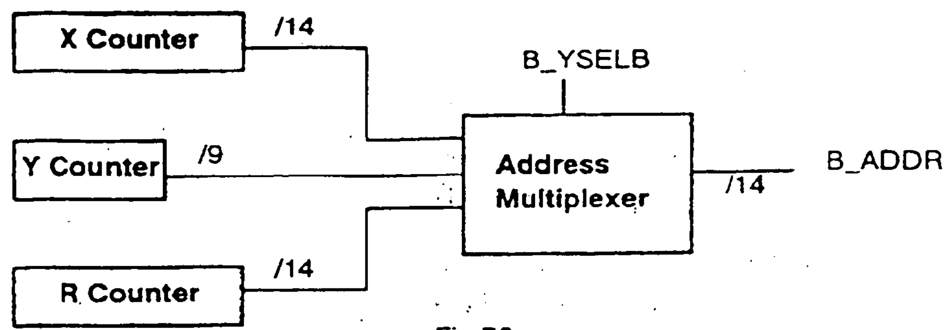


Fig 76

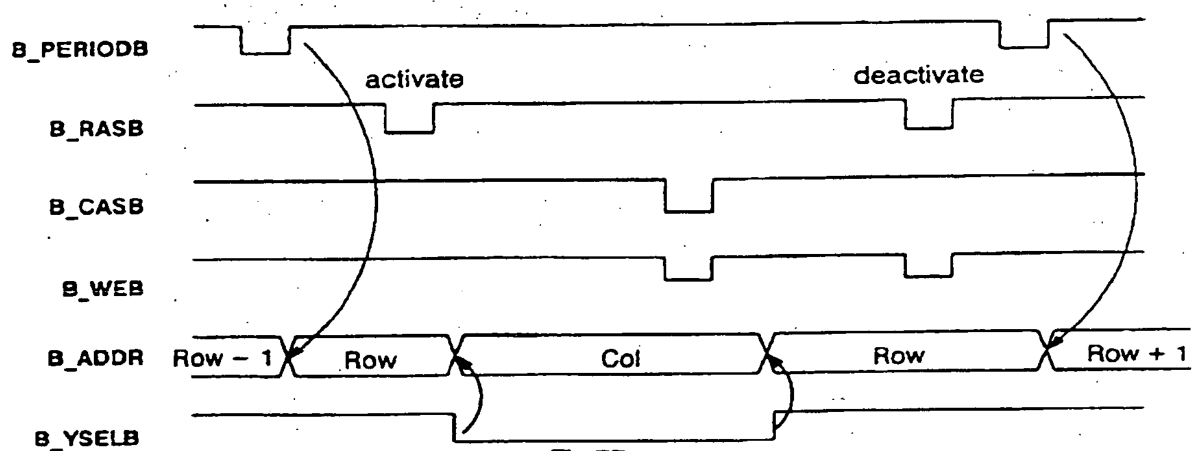


Fig 77

BA1	BA0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Int/Ext CLK	Sub Array	Output Enable	R1C	W1R	BI	Pdist	Ldist	Sdist	OV	Ymar	Xmar	Pause	Gross

For A0 to A3, '1' = Test of concerned enabled '0' = Test of concerned not enabled
For A4 = Overvoltage, 2 to 2 1/2 times VCC
For A5 to A10, '1' = Test of concerned enabled '0' = Test of concerned not enabled
For BA0, '1' indicate sub array selected. '0' indicate full array test.
For BA1, '1' indicate external clock. '0' indicate internal clock.
For A11, '1' indicate output enable, 0 indicate output not enabled. Only meaningful after BIST is done

Fig 78

	NORMAL	TLROMR	MON1	MON2	MON3
B_DQ(31)	High Z	RomBit 11	BA1	BA1	BA1
B_DQ(30)	High Z	RomBit 10	BA0	BA0	BA0
B_DQ(29)	High Z	RomBit 9	ADDR11	ADDR11	ADDR11
B_DQ(28)	High Z	RomBit 8	ADDR10	ADDR10	ADDR10
B_DQ(27)	High Z	RomBit 7	ADDR9	ADDR9	ADDR9
B_DQ(26)	High Z	RomBit 6	BIST_CMP	ADDR8	BIST_CMP ADDR8
B_DQ(25)	High Z	RomBit 5	B_DODD	ADDR7	B_DODD ADDR7
B_DQ(24)	High Z	RomBit 4	B_DEVEN	ADDR6	B_DEVEN ADDR6
B_DQ(23)	High Z	RomBit 3	B_RASB	ADDR5	B_RASB ADDR5
B_DQ(22)	High Z	RomBit 2	B_CASB	ADDR4	B_CASB ADDR4
B_DQ(21)	High Z	RomBit 1	B_WB	ADDR3	B_WB ADDR3
B_DQ(20)	High Z	RomBit 0	B_CLK	ADDR2	B_CLK ADDR2
B_DQ(19)	High Z	Low	ADDR1	ADDR1	ADDR1
B_DQ(18)	High Z	Low	ADDR0	ADDR0	ADDR0
B_DQ(17)	PASSFAIL	High Z	High Z	High Z	High Z
B_DQ(16)	B_DONE	High Z	High Z	High Z	High Z

Fig 79

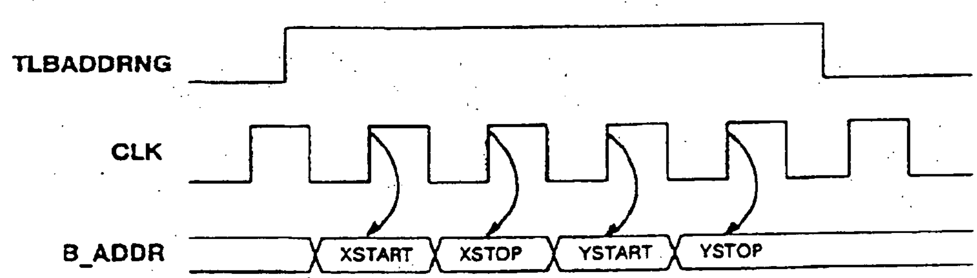


Fig 80

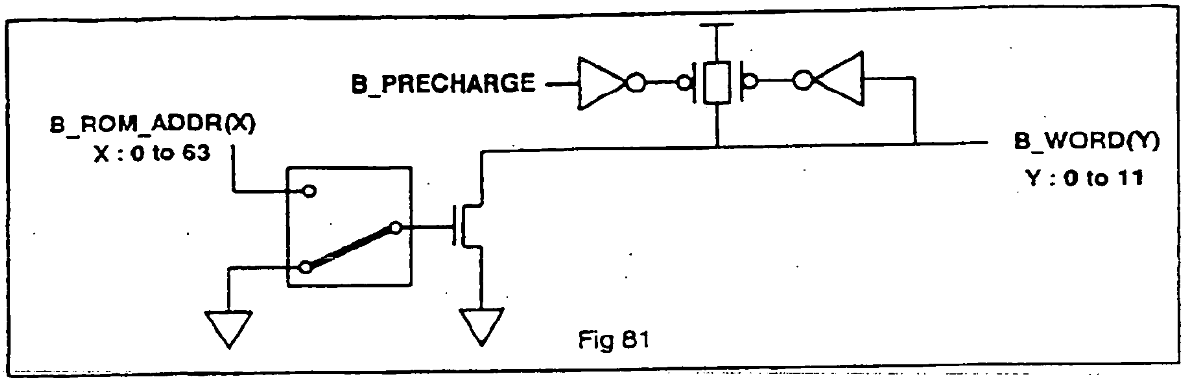


Fig 81

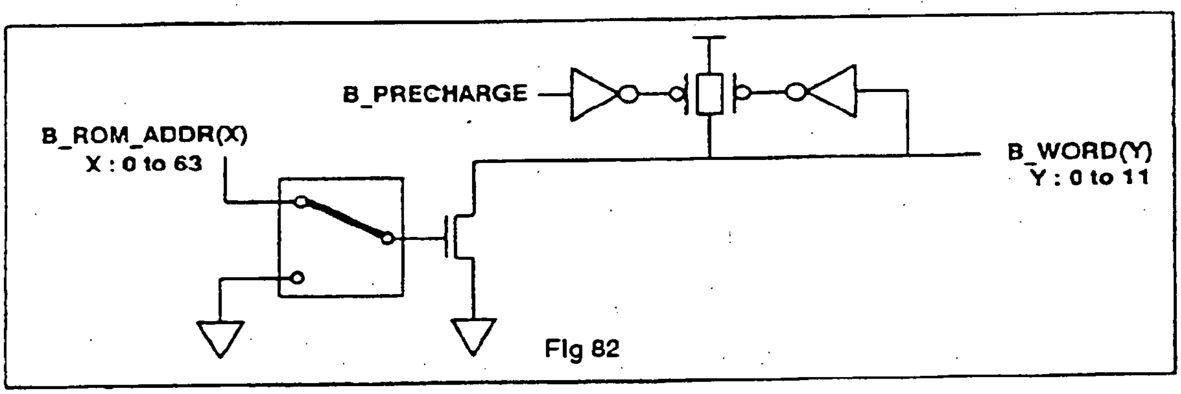


Fig 82

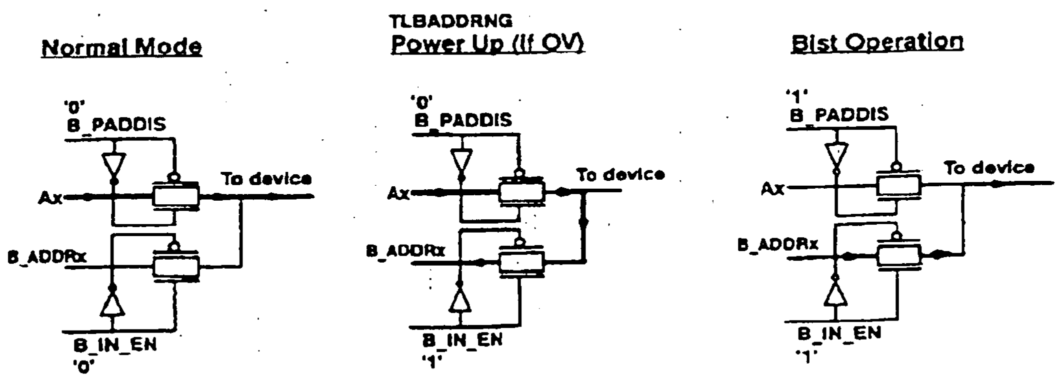


Fig 83

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 30 2975

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	IEICE TRANSACTIONS ON ELECTRONICS, vol. e78-c, no. 7, July 1995, TOKYO JP, pages 838-844, XP000528815 KOIKE H. ET AL.: "BIST Circuit using microprogram ROM for LSI memories" * the whole document *	1,2,6	G11C29/00 G06F11/27 G01R31/3187
Y		3-5	
Y	US 5 327 363 A (AKIYAMA) * abstract; claims 1,2; figures 1,2 *	3-5	
A	PROCEEDINGS INTERNATIONAL TEST CONFERENCE 1990, 10 September 1990 - 14 October 1990, WASHINGTON DC. USA, pages 815-822, XP000203754 KOIKE H. ET AL.: "A BIST scheme using microprogram ROM for large capacity memories" * the whole document *	1-6	
A	COMPUTER, vol. 23, no. 10, October 1990, LONG BEACH US, pages 45-56, XP000170952 FRANKLIN M. ET AL.: "Built-in self-Testing of Random-Access memories" * page 51 - page 55 *	1-6	TECHNICAL FIELDS SEARCHED (Int.Cl.6) G11C G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	11 August 1997	Sarasua Garcia, L

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

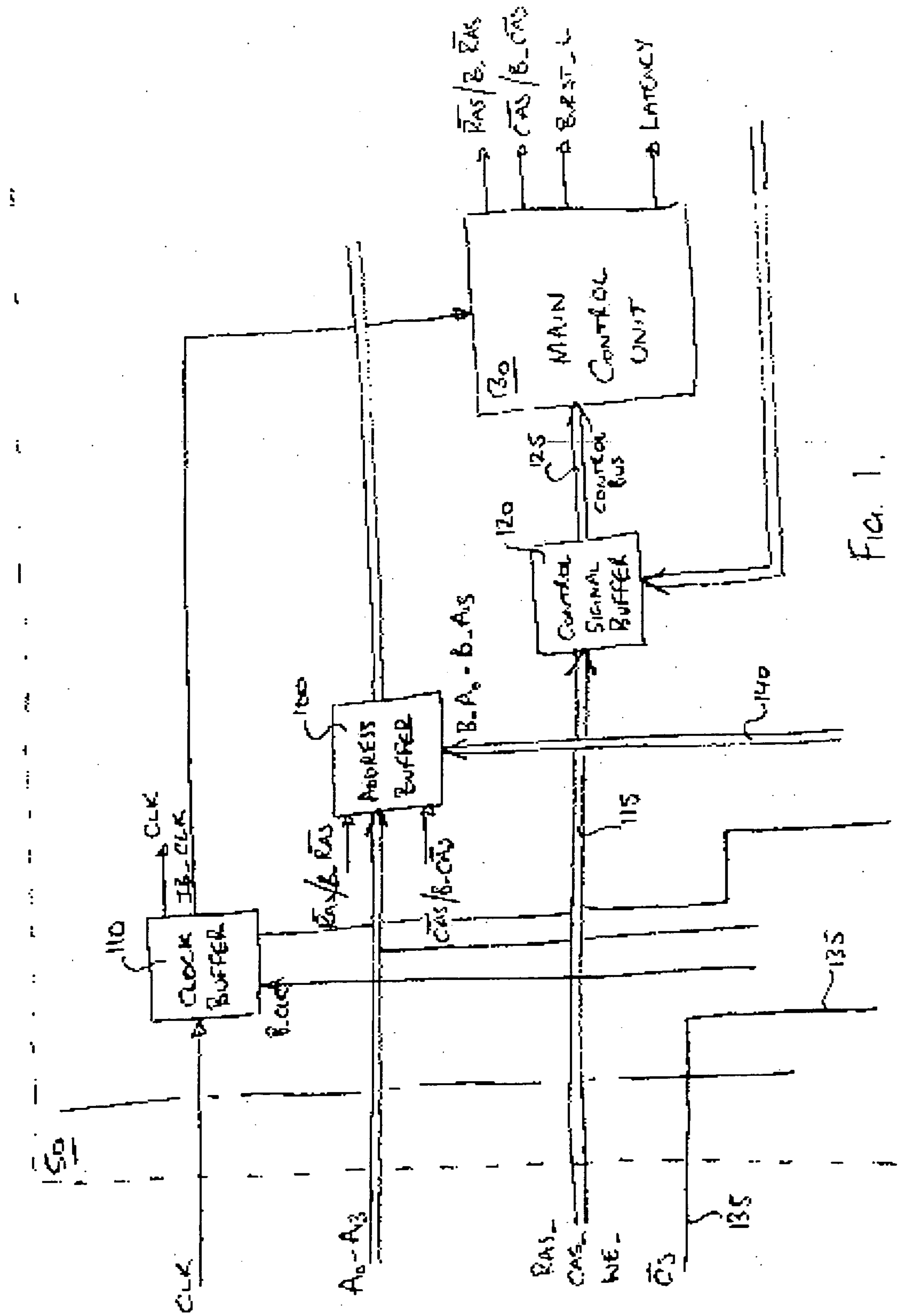

MAIN CONTROL UNIT
CLOCK BUFFER
ADDRESS BUFFER
CONTROL SIGNAL BUFFER
CONTROL BUS
LATENCY
CLK
110
100
120
125
115
140
135

FIG. 1.

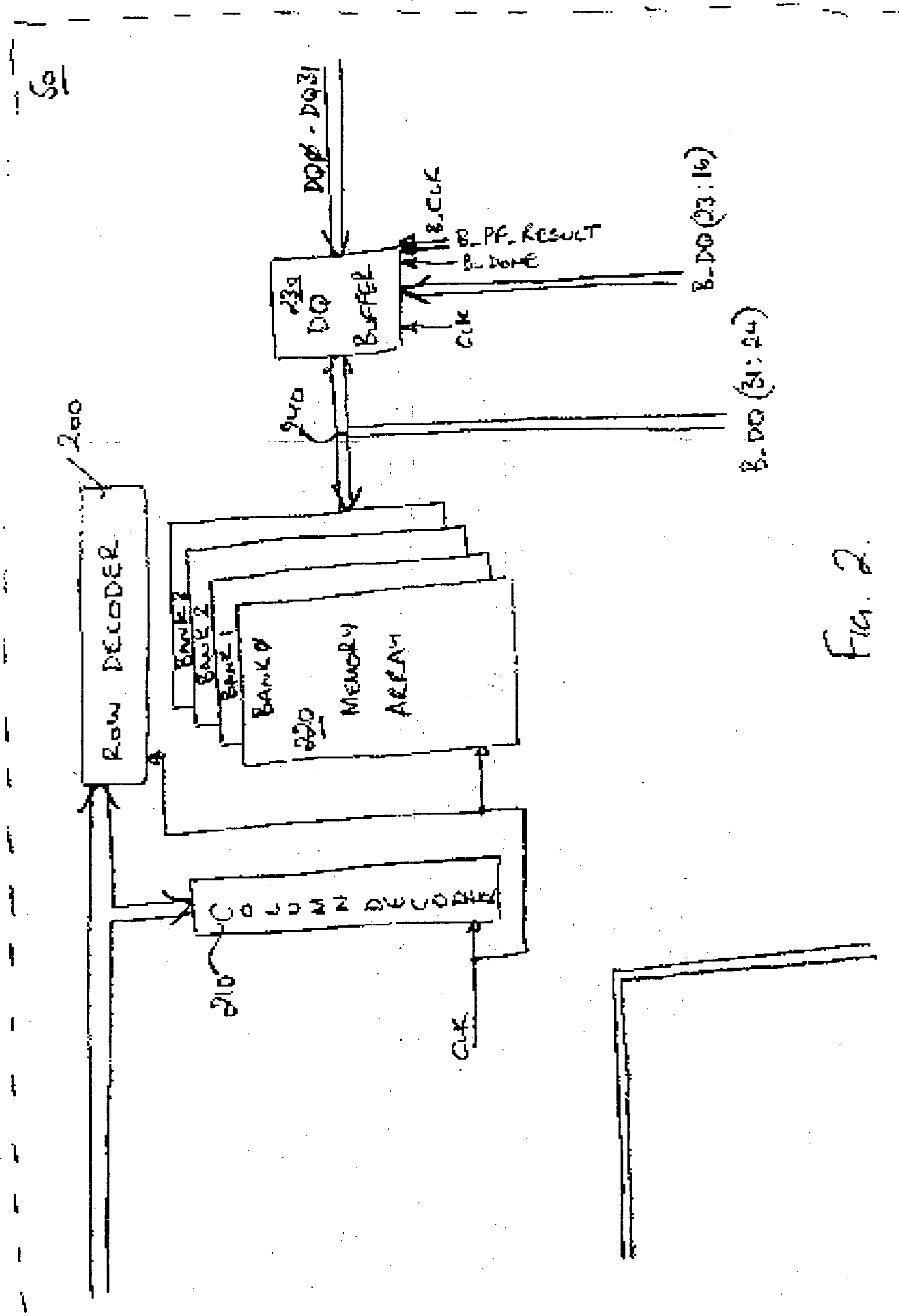
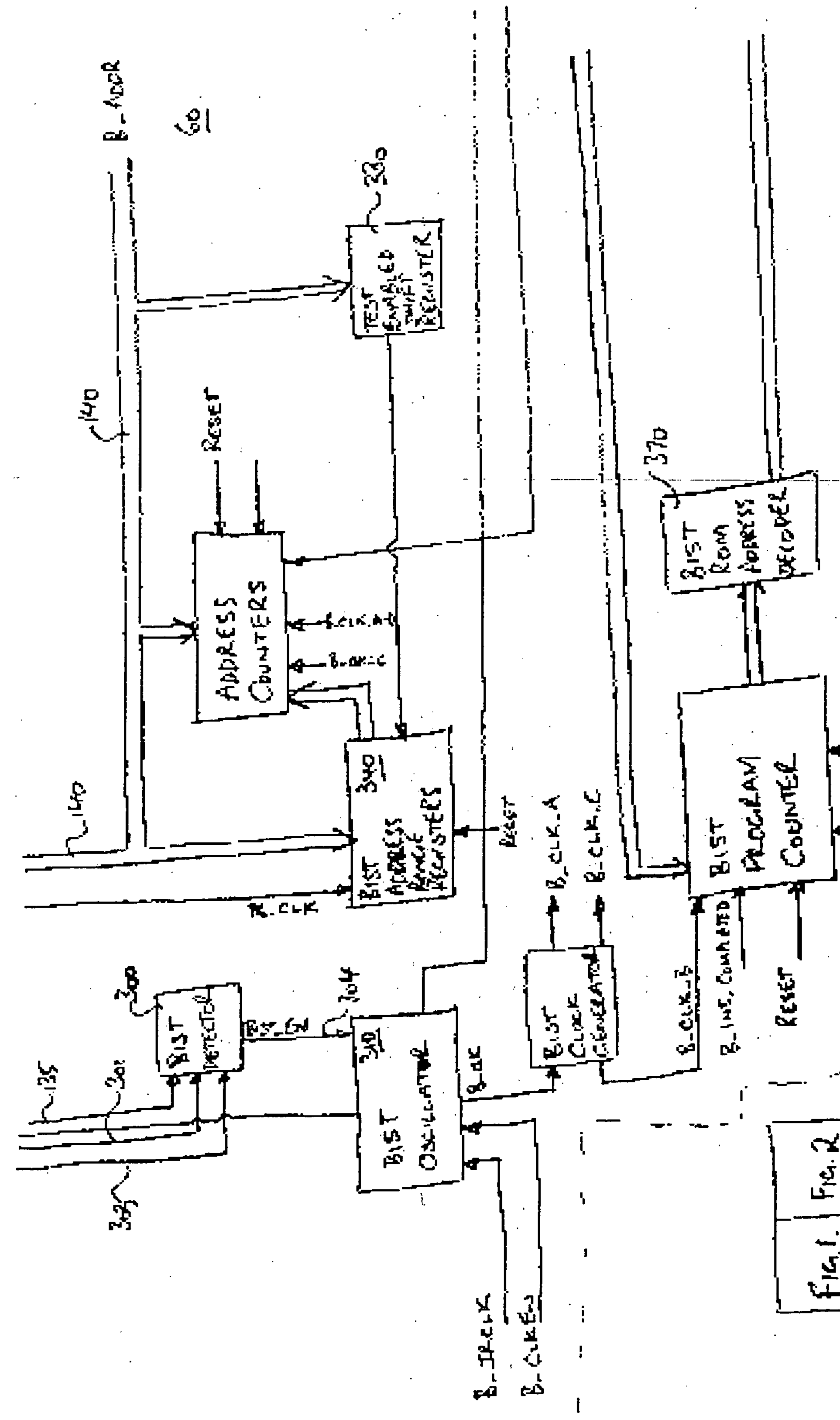

DQ0 - DQ31
B_CLK
B_PF_RESULT
B_DONE
B_DQ (23:16)
230
DQ
BUFFER
CLK
240
B_DQ (31:24)
200
ROW DECODER
BANK 3
BANK 2
BANK 1
BANK 0
220
MEMORY
ARRAY
210
CLK

FIG. 2



B_ADDR
60
330
TEST ENABLED INPUT REGISTER
140
RESET
370
ADDRESS COUNTERS
B_CLK_A
B_CLK_C
BIST ROM ADDRESS DECODER
340
BIST ADDRESS RANGE REGISTERS
RESET
B_CLK
JUMP
BIST PROGRAM COUNTER
INC_CTR
300
BIST DETECTOR
BIST_EN
304
310
BIST OSCILLATOR
B_CLK
BIST CLOCK GENERATOR
B_CLK_A
B_CLK_C
B_CLK_B
B_INS_COMPLETED
RESET
135
301
302
B_CLKEN

FIG. 3

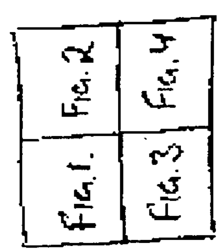


Fig.1
Fig.2
Fig.3
Fig.4

FIG. 5

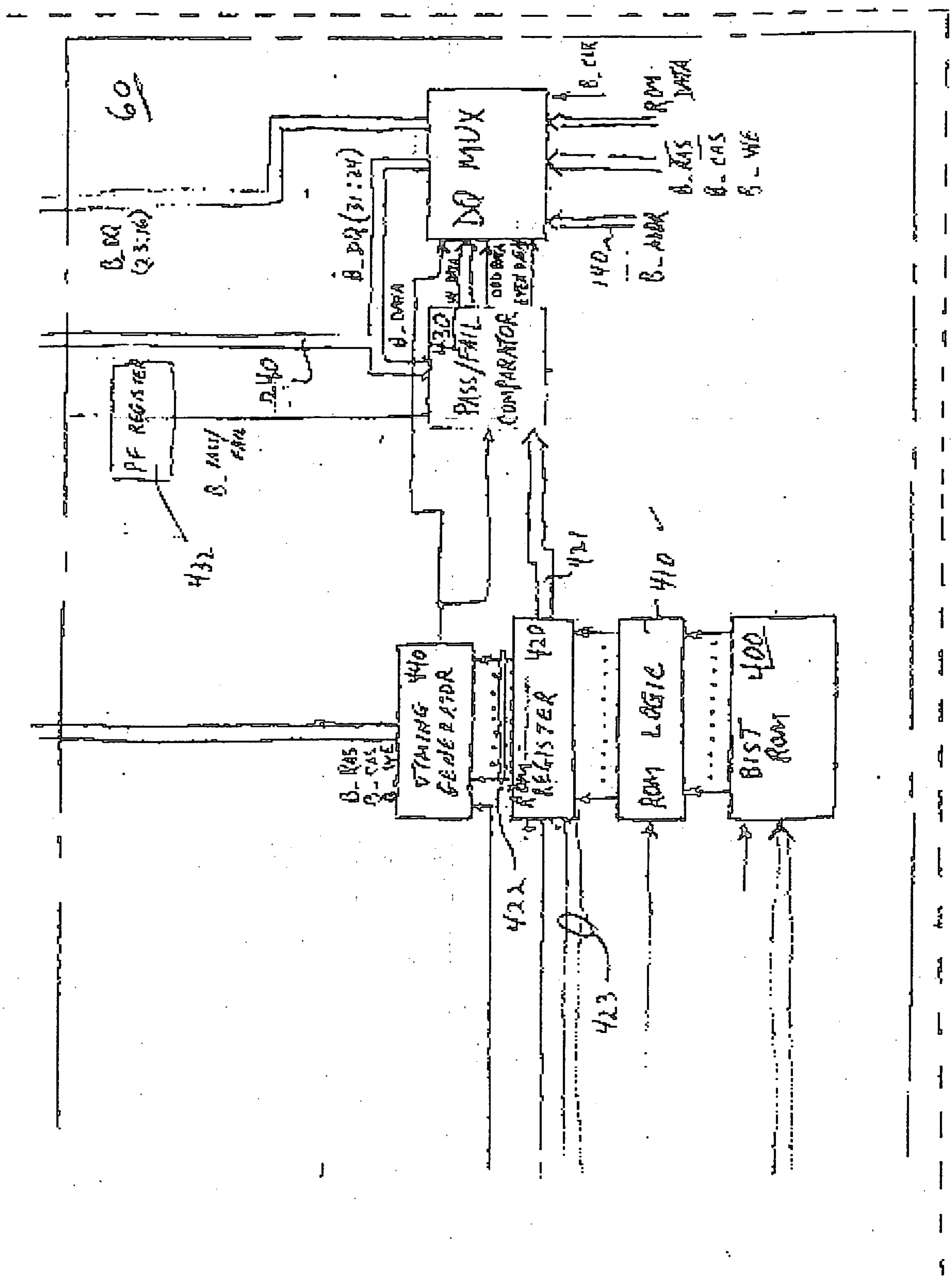

60
DQ MUX
B_CAR
ROM DATA
B_RAS
B_CAS
B_WE
B_ADDR
140
B_DQ (23:16)
B_DQ(31:24)
B_DATA
W DATA
ODD DATA
EVEN DATA
430
PASS/FAIL
COMPARATOR
240
PF REGISTER
B_PASS/FAIL
432
421
410
TIMING 440
GENERATOR
ROM REGISTER 420
ROM LOGIC
BIST ROM 400
B_RAS
B_CAS
B_WE
422
423

FIG. 4

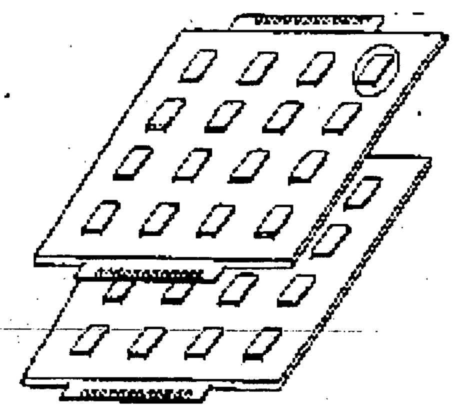


FIG. 6

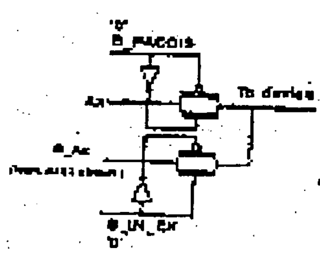


FIG. 13A

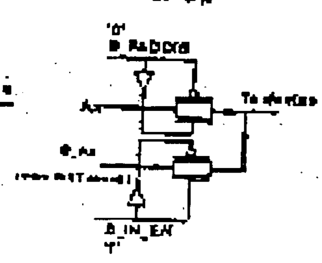


FIG. 13B

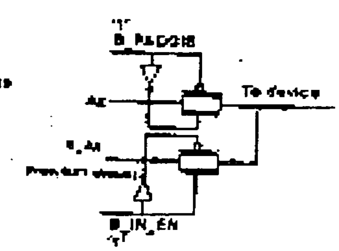


FIG. 13C

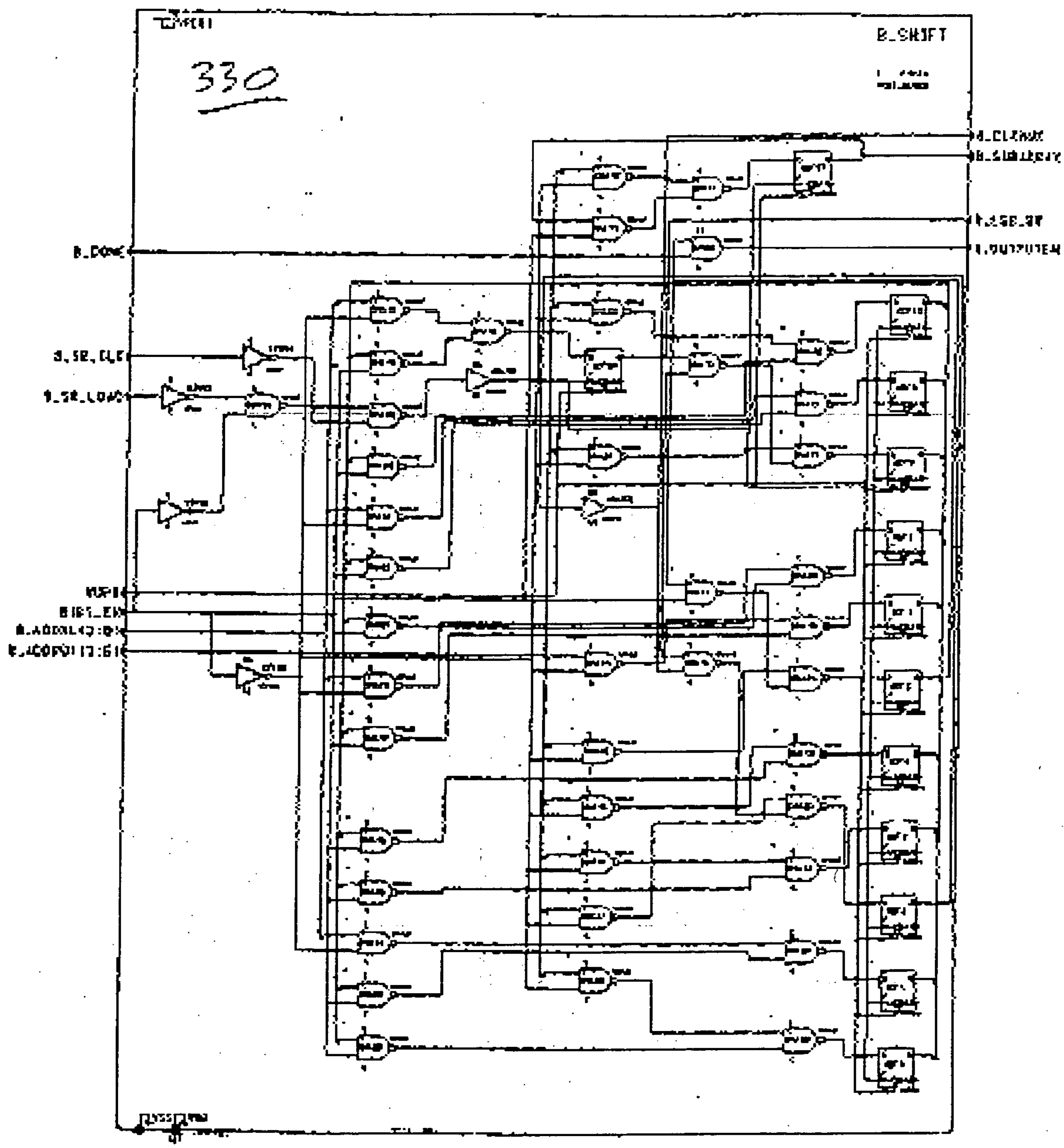


FIG. 7

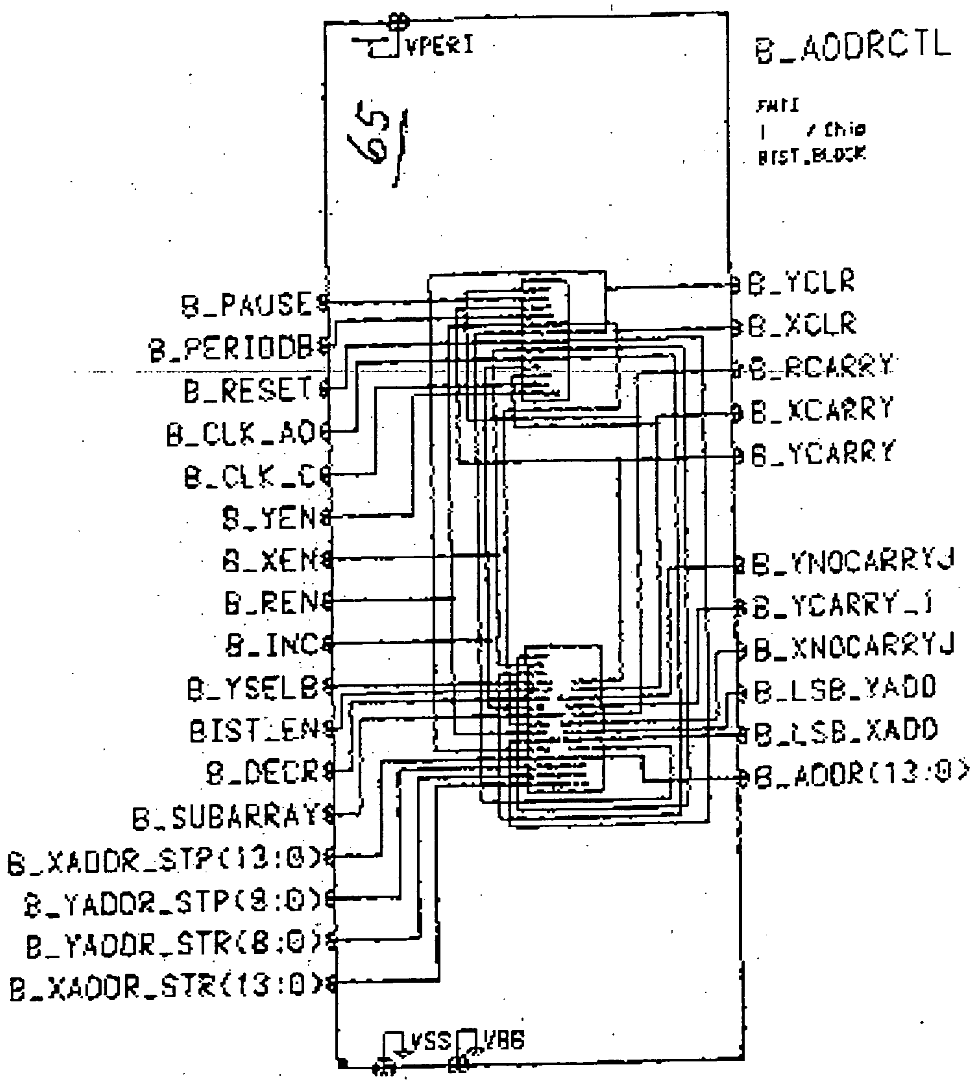

VPERI
B_ADDRCTL
FATI
/ Chip
BIST_BLOCK
65
B_PAUSE
B_PERIODB
B_RESET
B_CLK_AD
B_CLK_C
B_YEN
B_XEN
B_REN
B_INC
B_YSELB
BIST_EN
B_DECR
B_SUBARRAY
B_XADDR_STP(13:0)
B_YADDR_STP(8:0)
B_YADDR_STR(8:0)
B_XADDR_STR(13:0)
B_YCLR
B_XCLR
B_RCARRY
B_XCARRY
B_YCARRY
B_YNOCARRYJ
B_YCARRY_J
B_XNOCARRYJ
B_LSB_YADD
B_LSB_XADD
B_ADDR(13:0)
VSS
VBB

FIG. 8

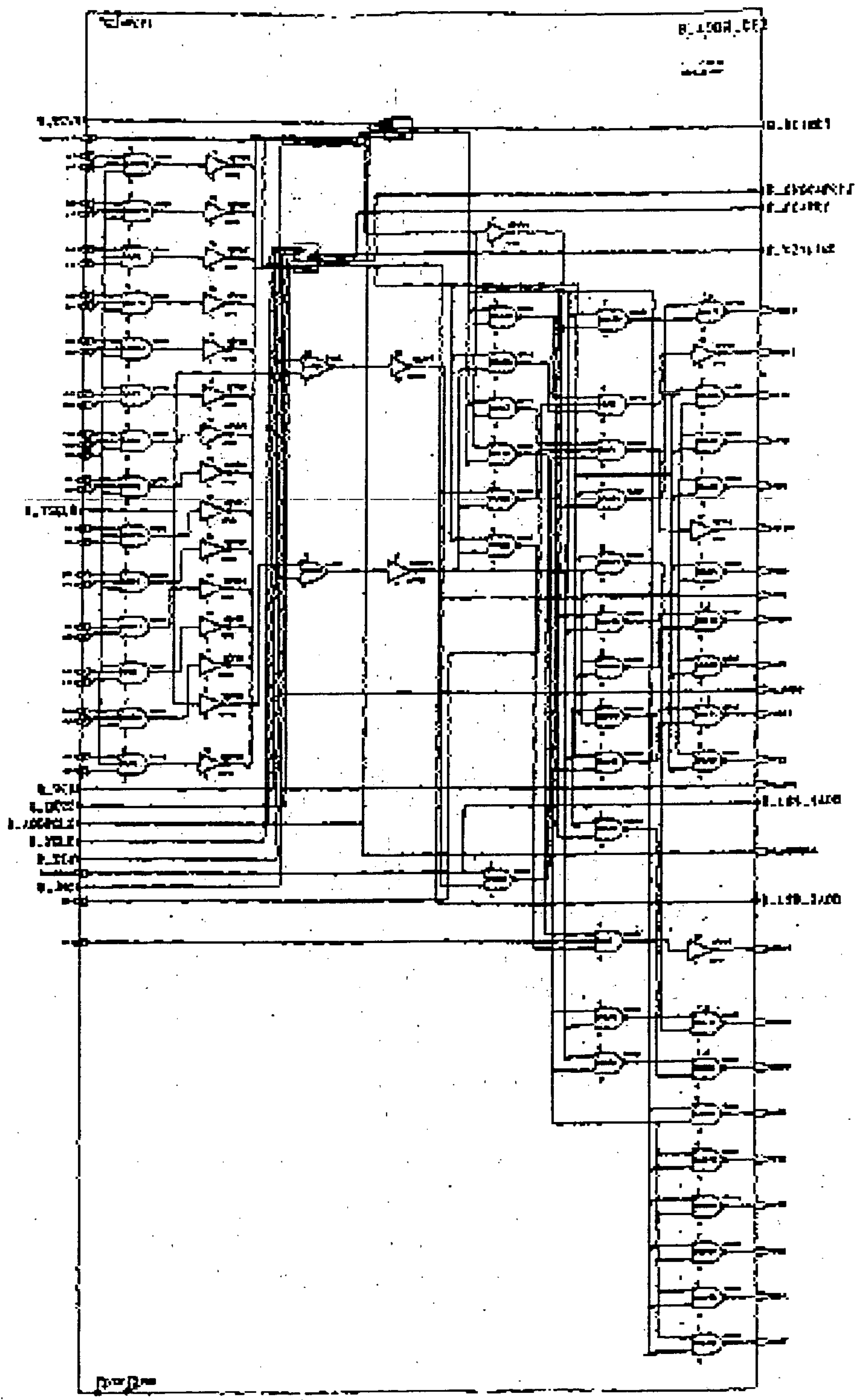


FIG. 9

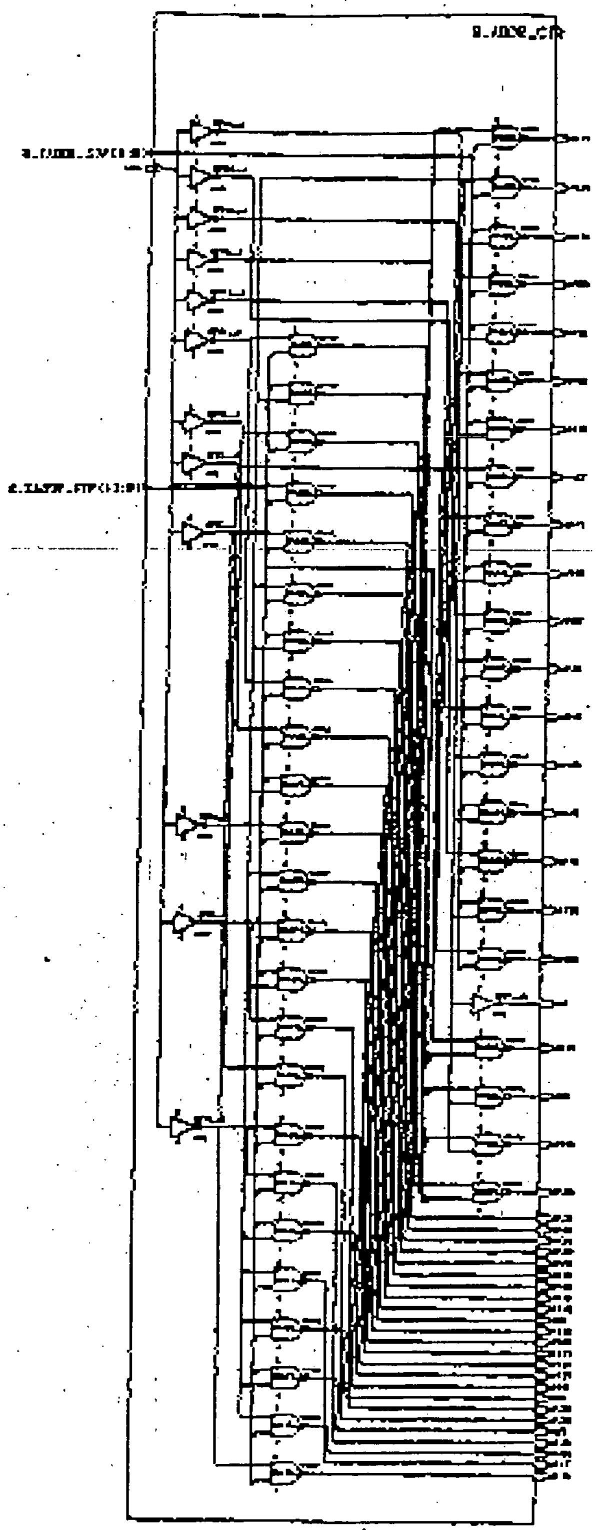


FIG. 10

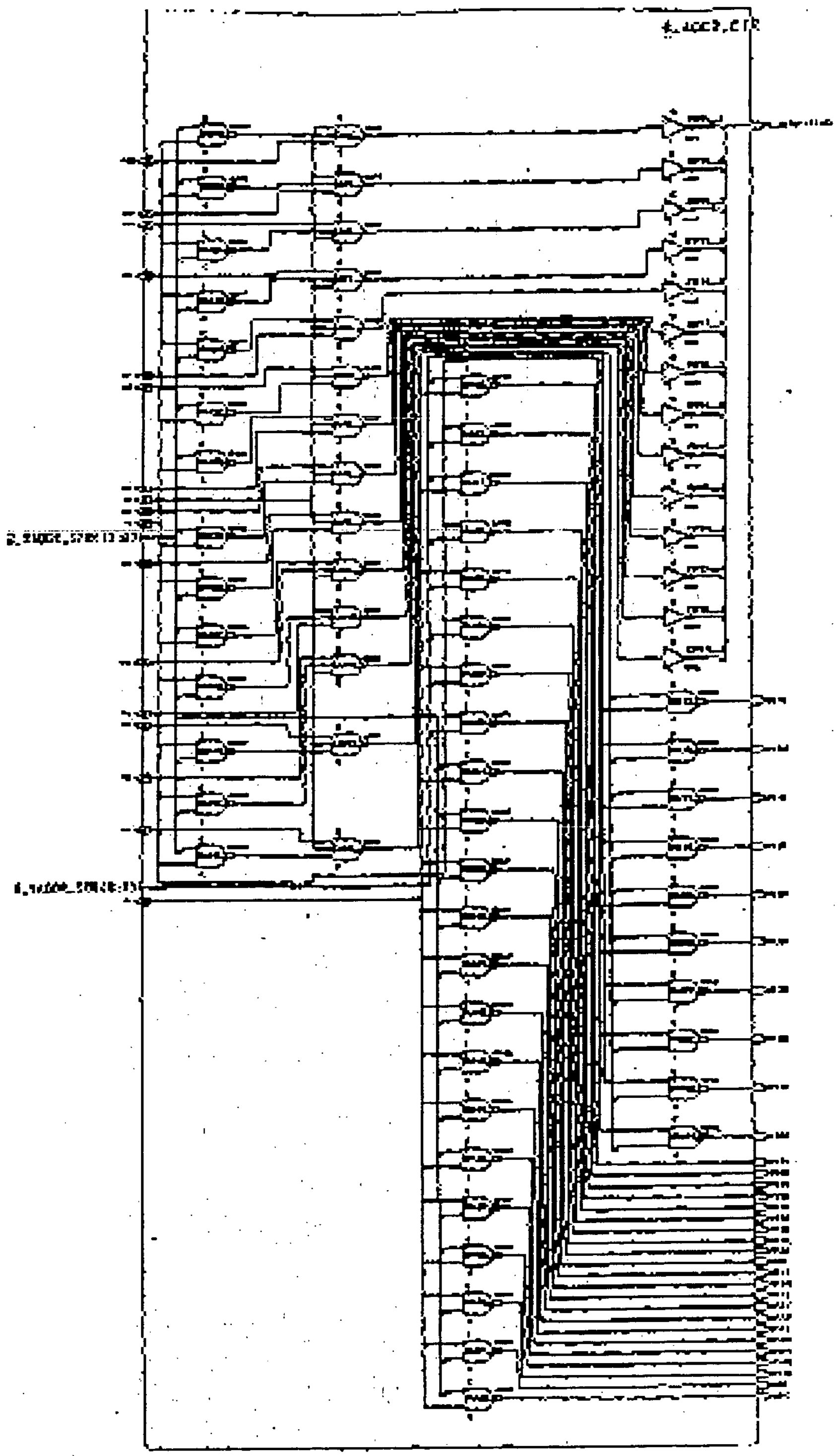


FIG. 11

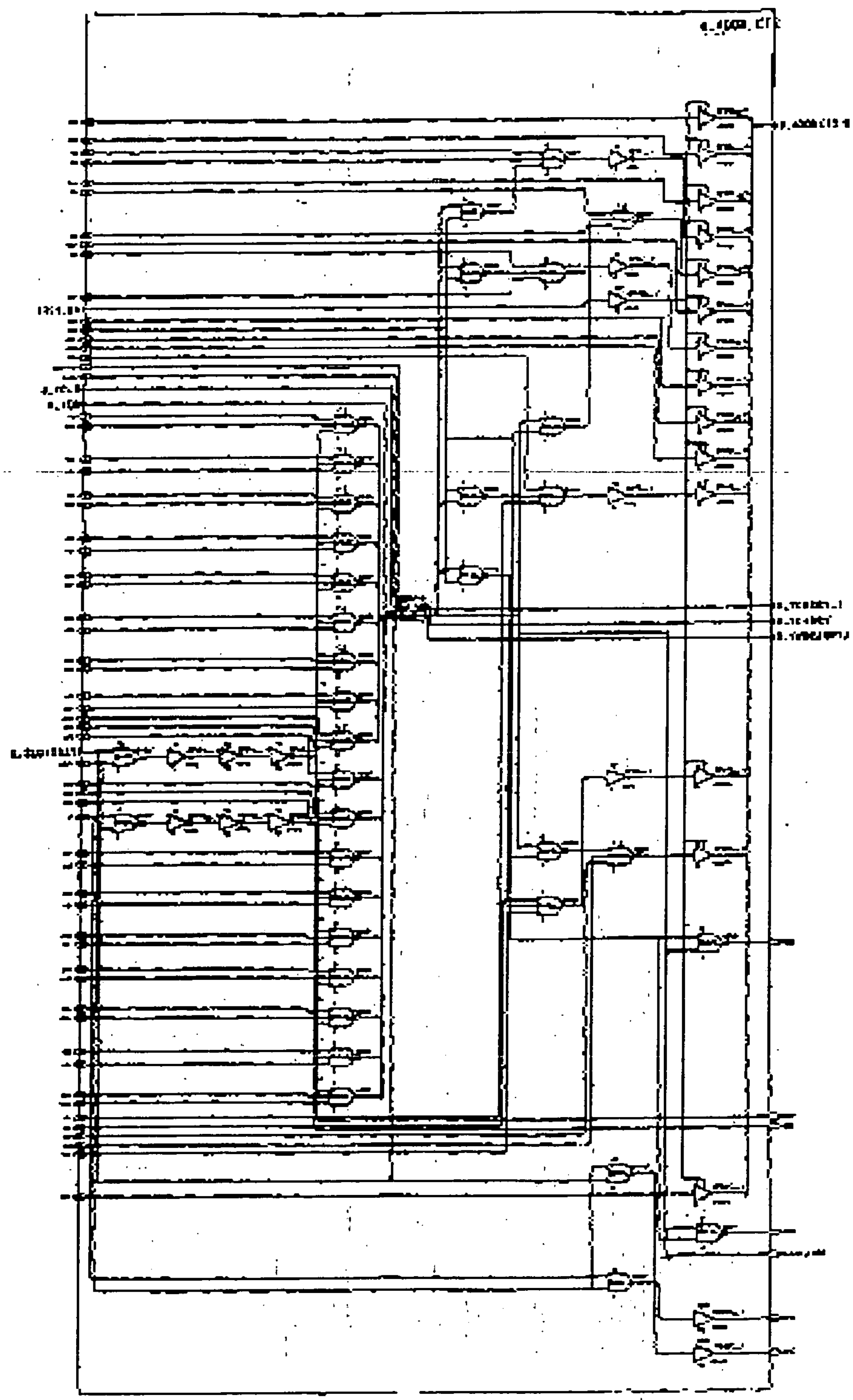


FIG. 12

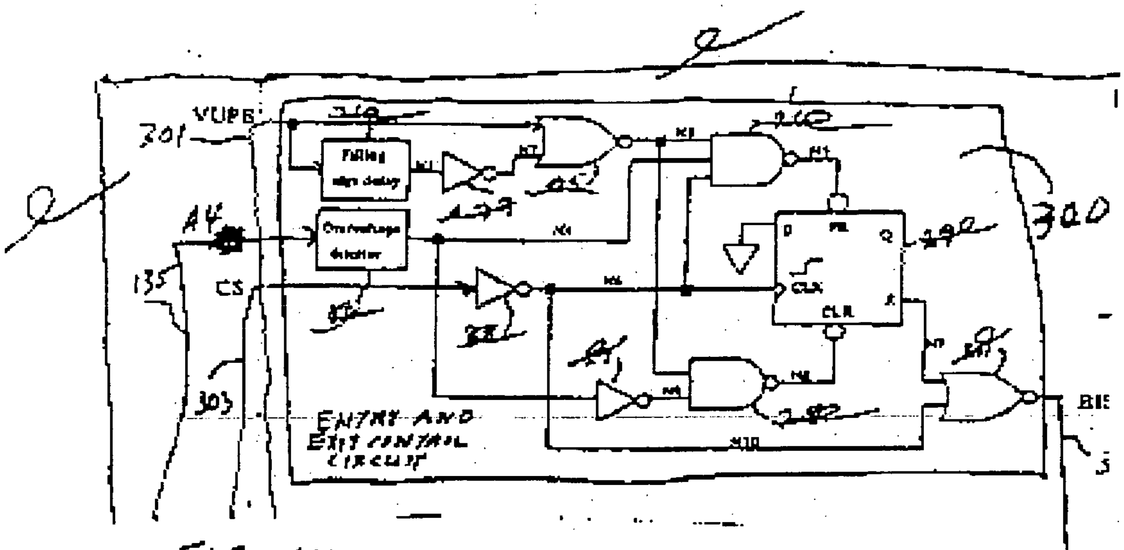


FIG. 14

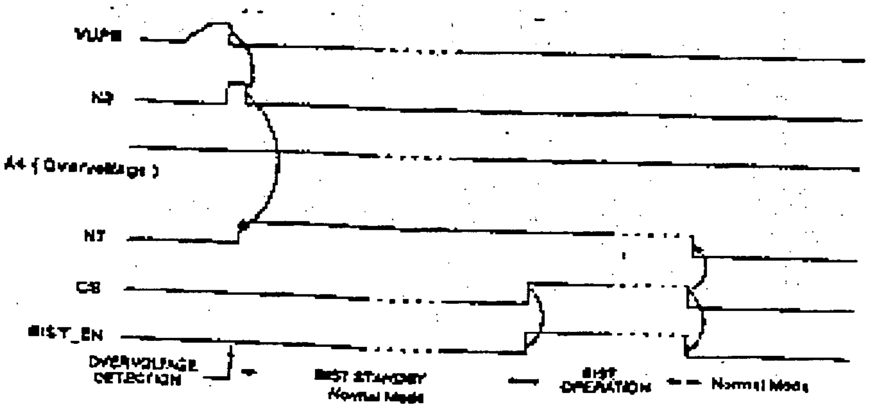


Figure 15

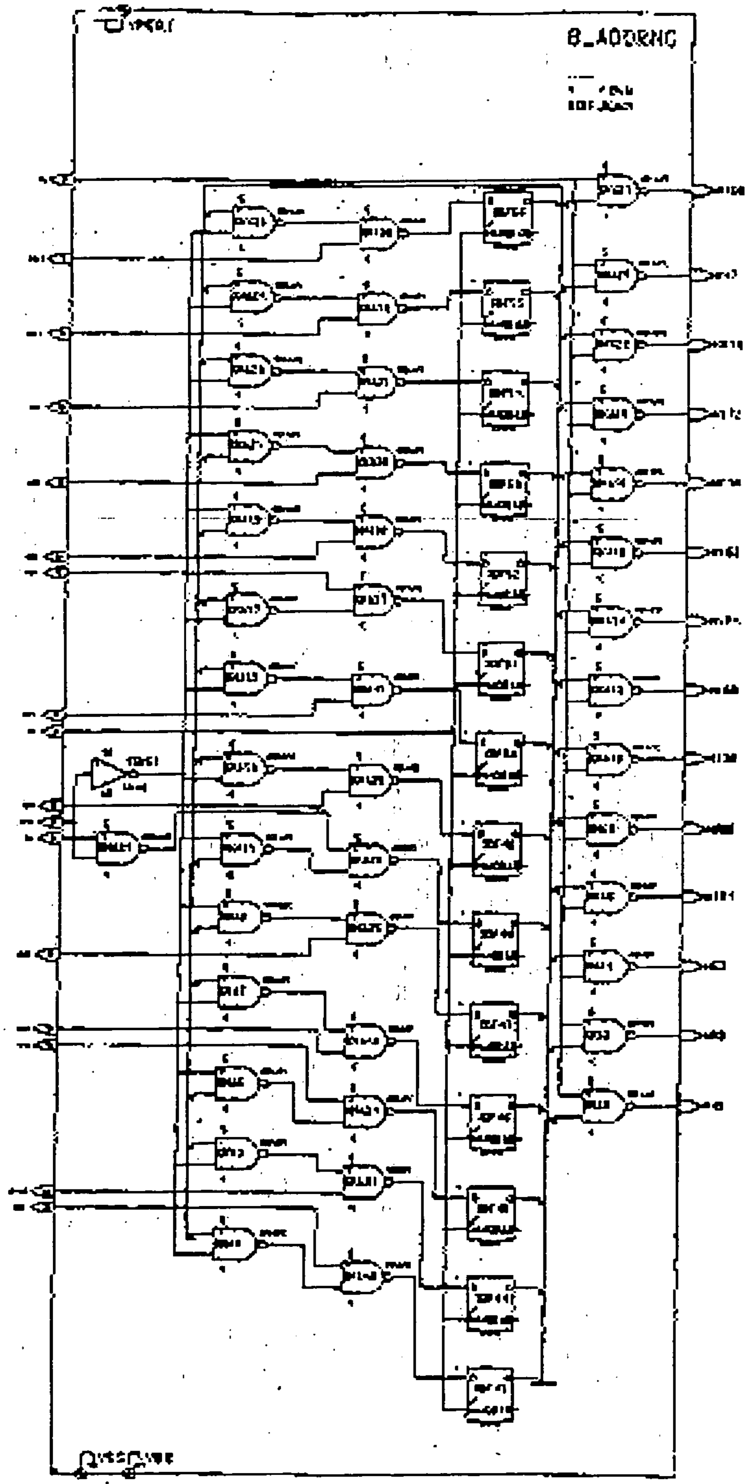


FIG. 16

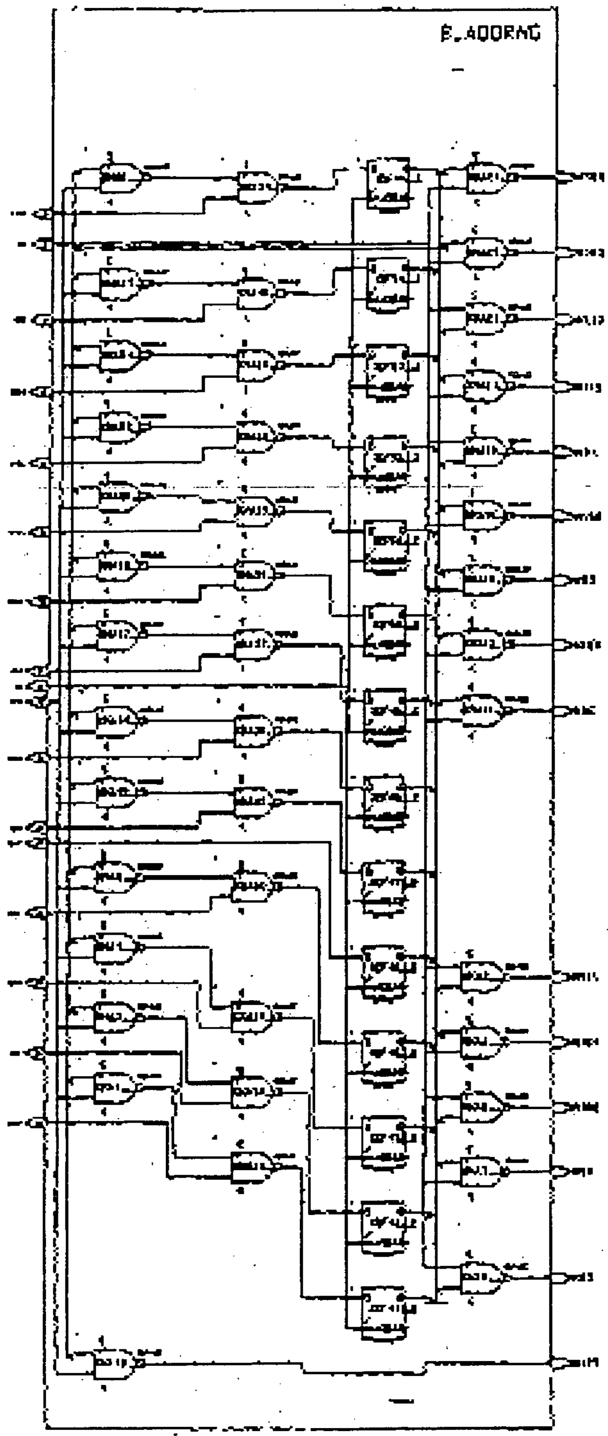


FIG. 17

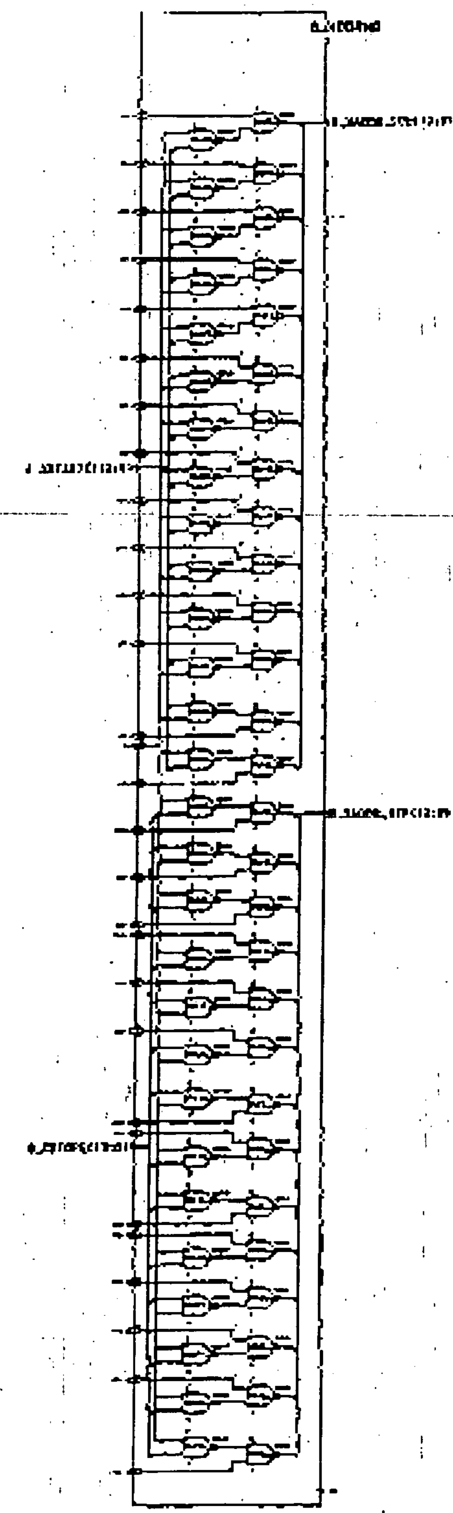
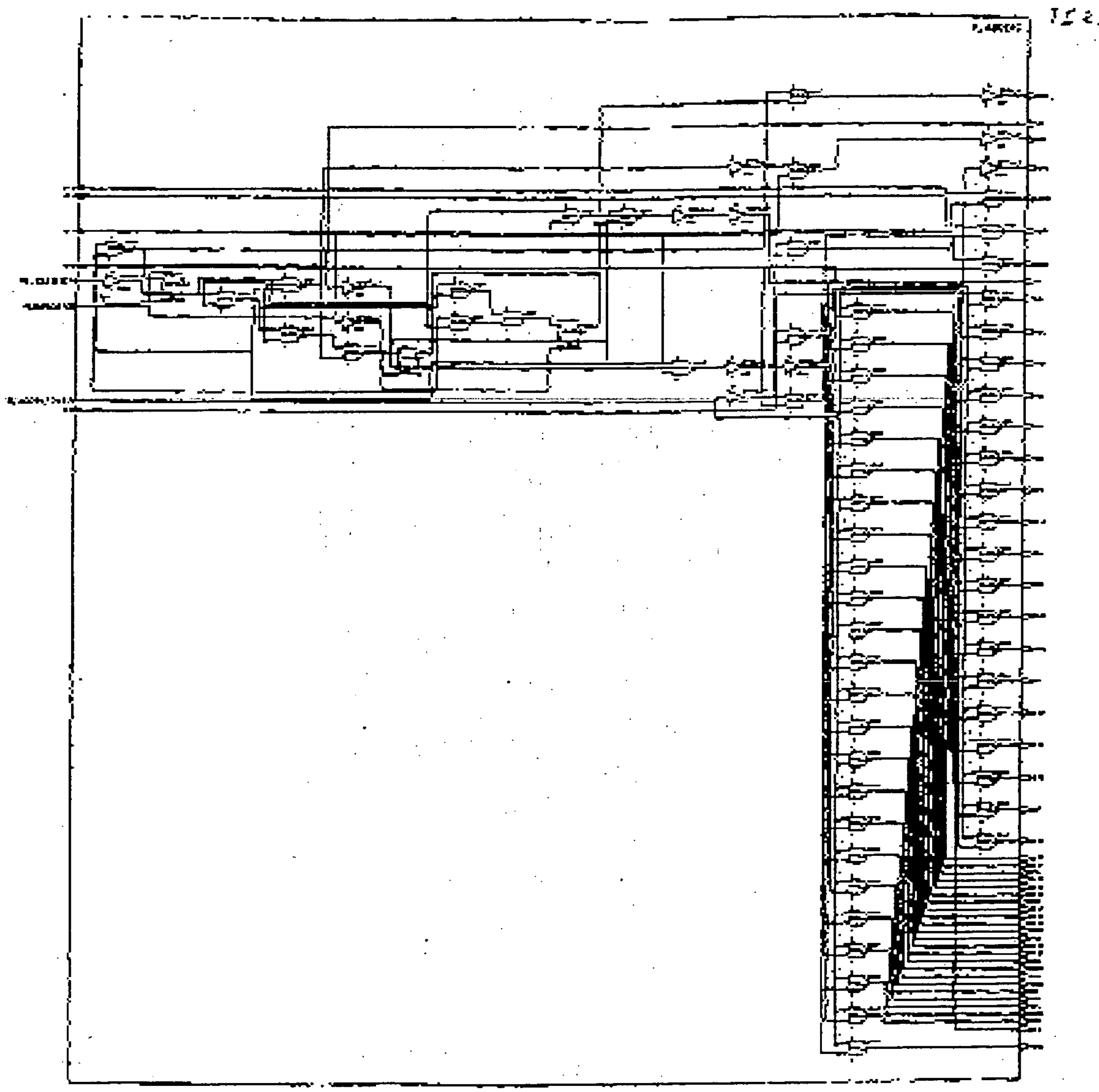


FIG. 1E



TI 23011

FIG. 19

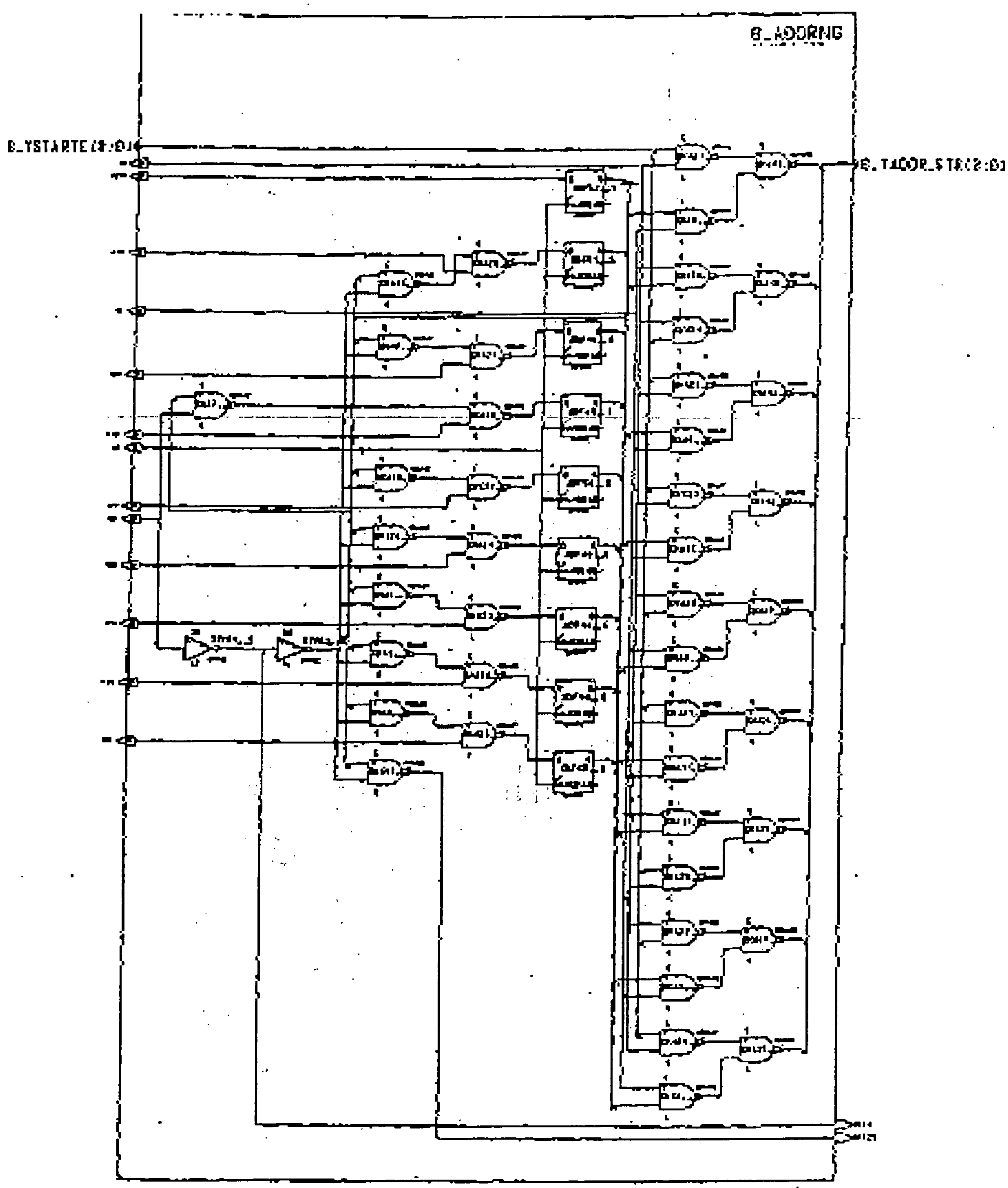


FIG. 20

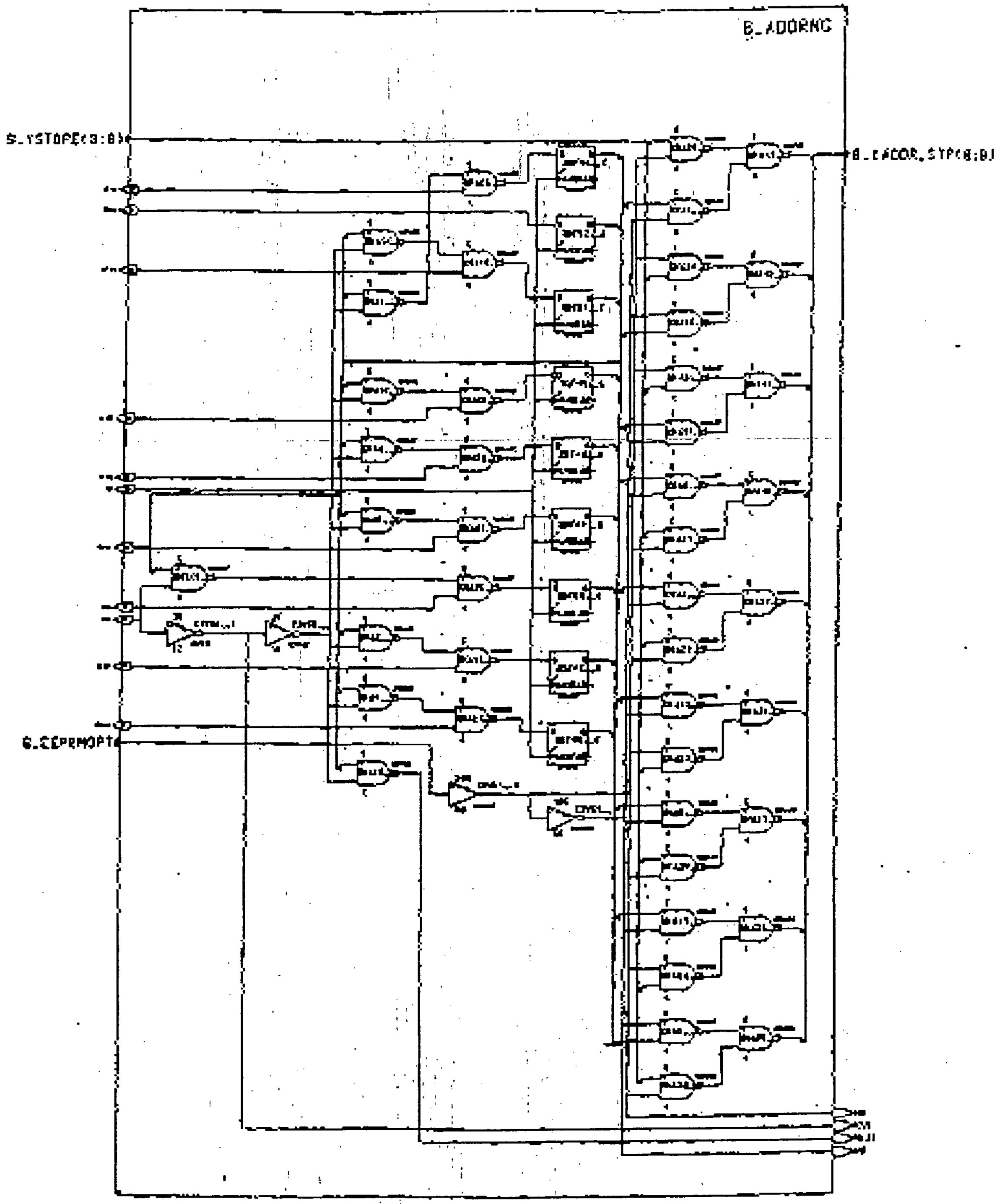


FIG. 21

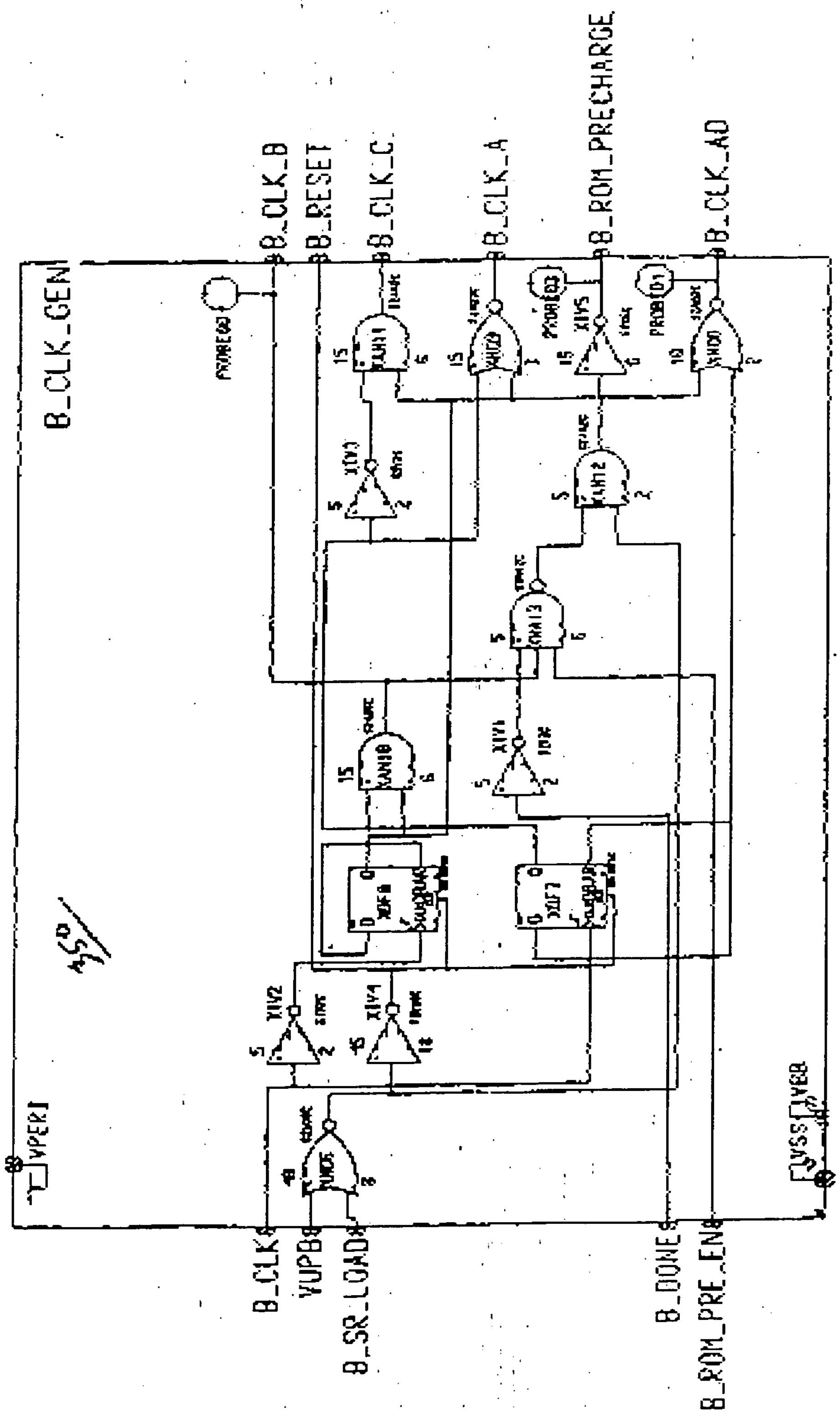

B_CLK_GEN
B_CLK_B
B_RESET
B_CLK_C
B_CLK_A
B_ROM_PRECHARGE
B_CLK_AD
VPERI
B_CLK
VUPB
B_SR_LOAD
B_DONE
B_ROM_PRE_EN
VSS
VBB

FIG. 22

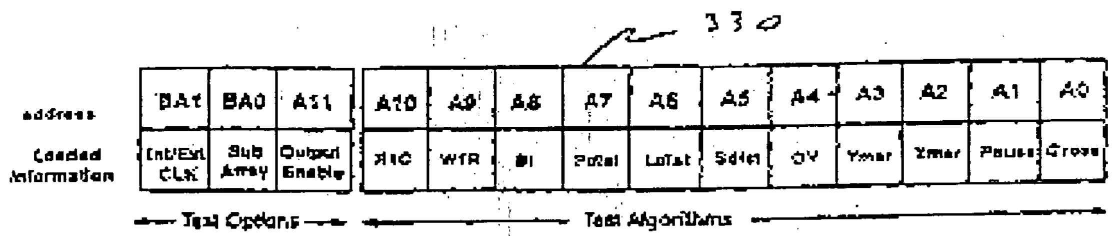

330
address
Loaded Information
BA1
BA0
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
Int/Ext CLK
Sub Array
Output Enable
WTR
BI
LoTst
OV
Ymar
Xmar
Pause
Cross
Test Options
Test Algorithms

FIG. 23

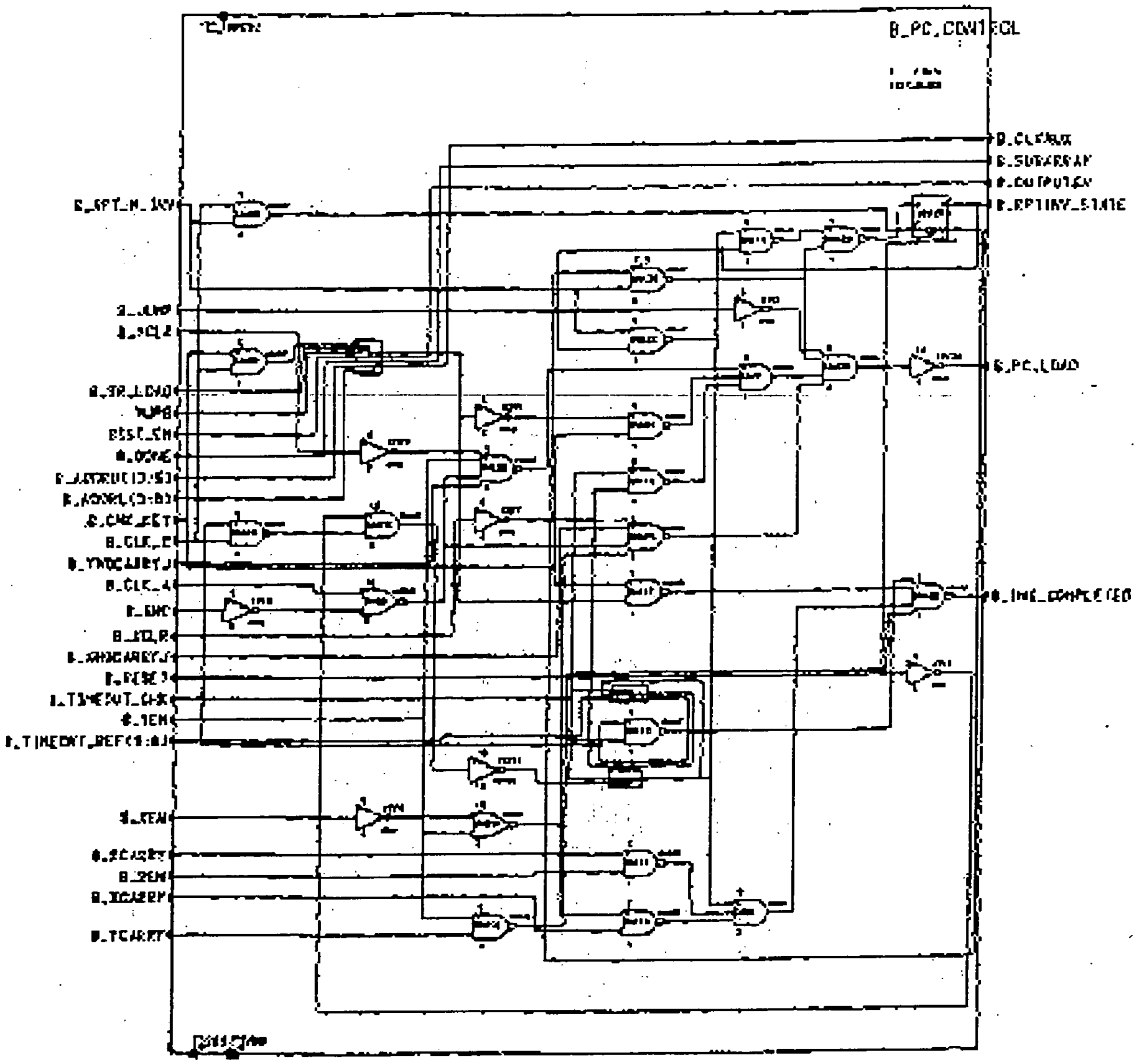


FIG. 24

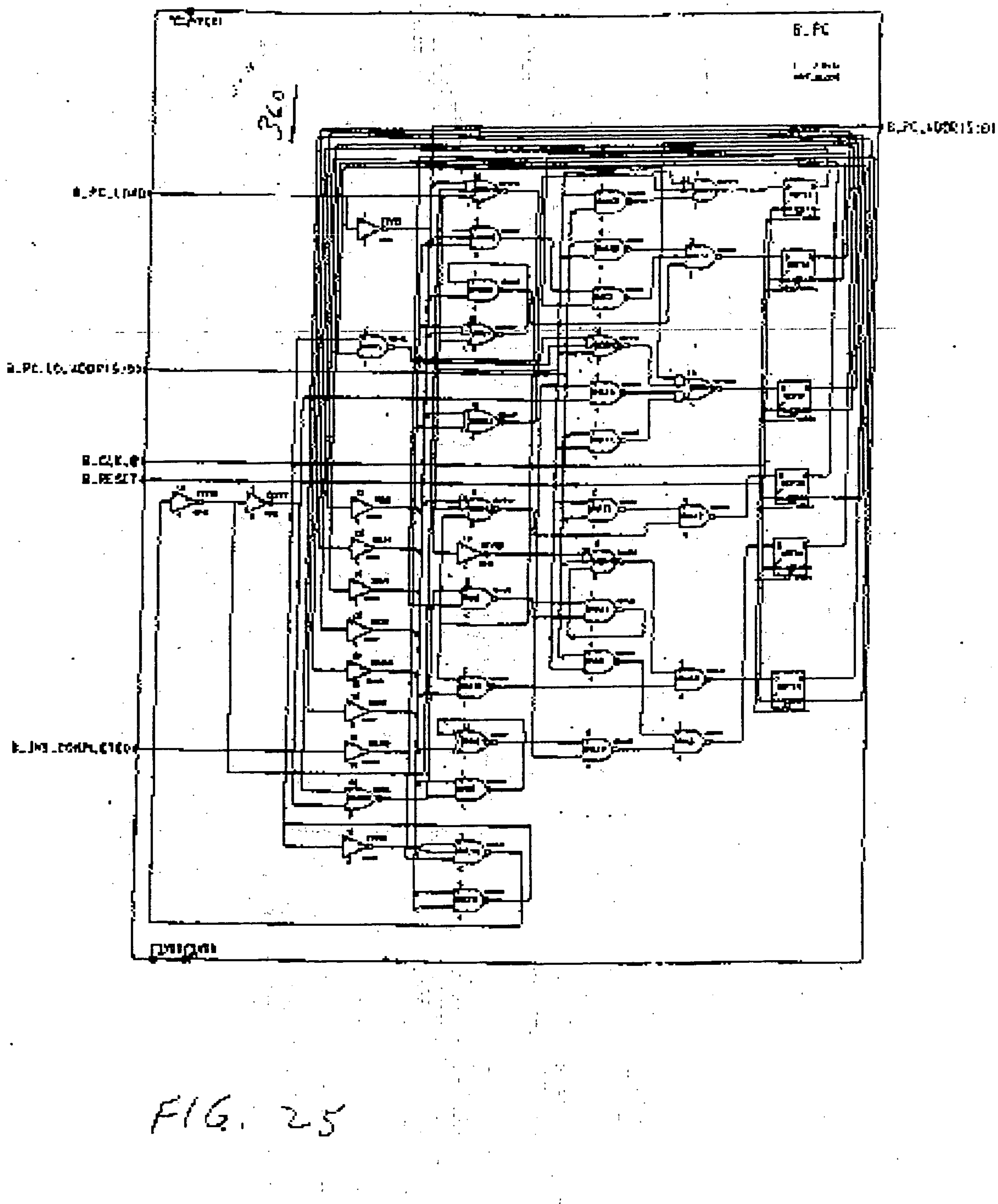


FIG. 25

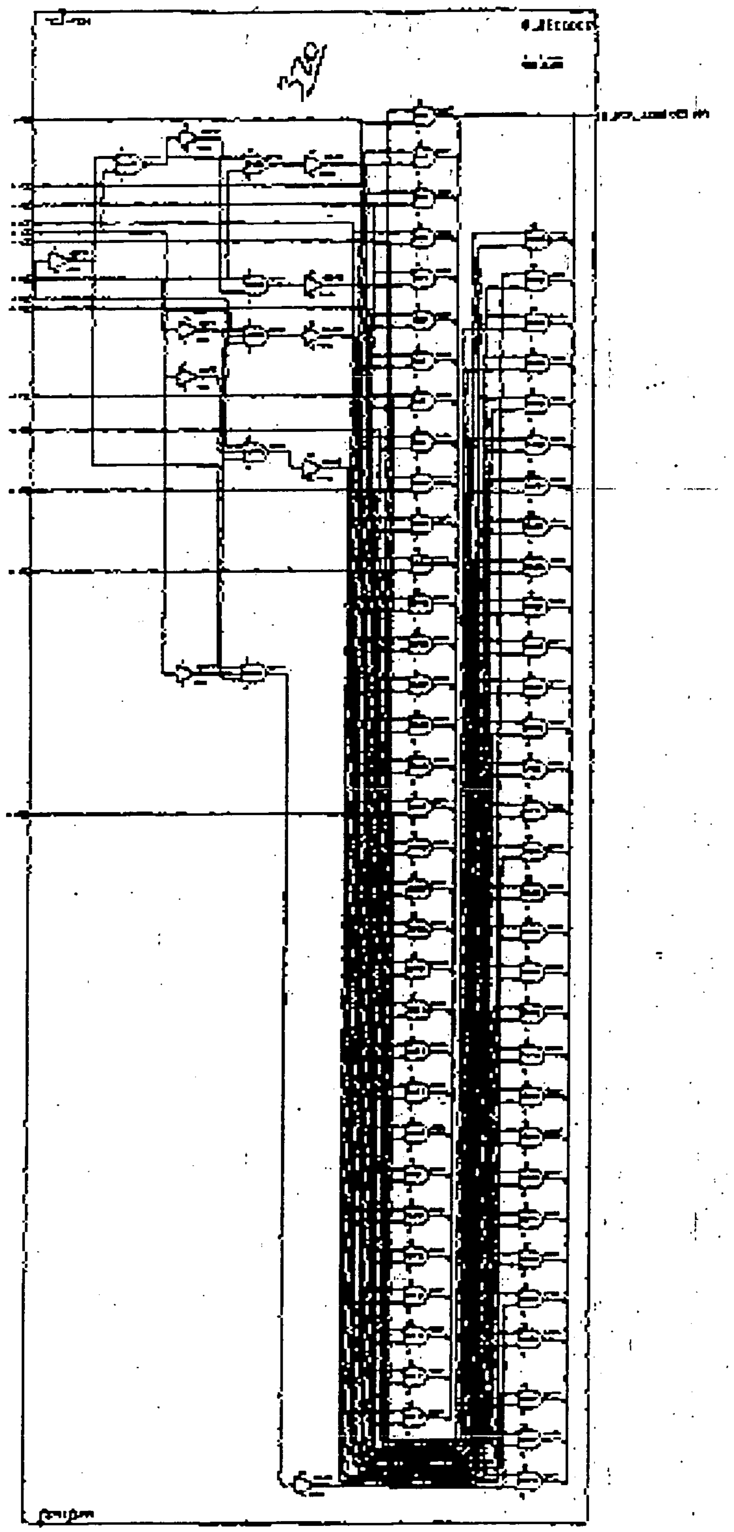


FIG. 26

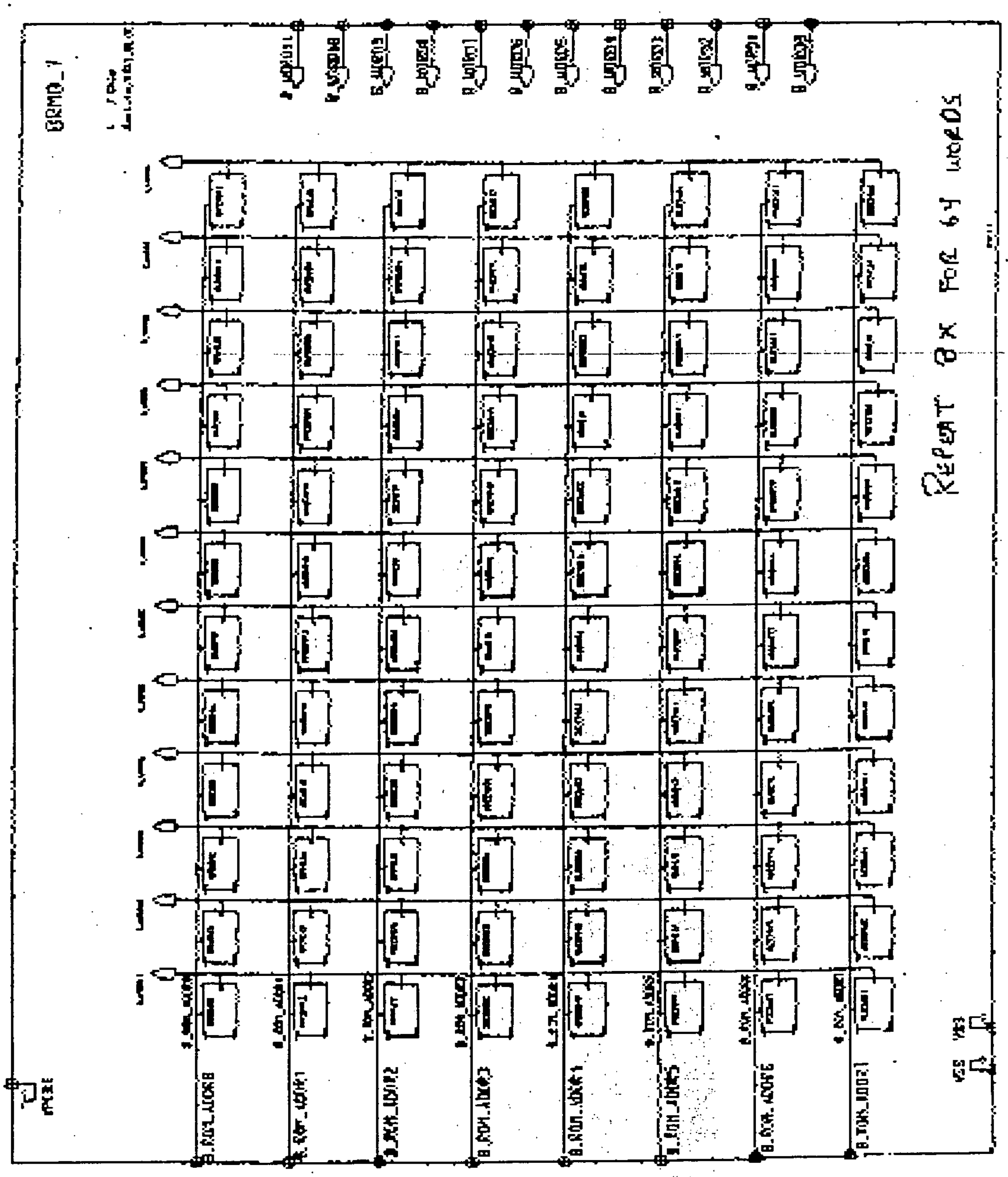


FIG. 27

256M BIST ALGORITHM STORAGE CONFIGURATION

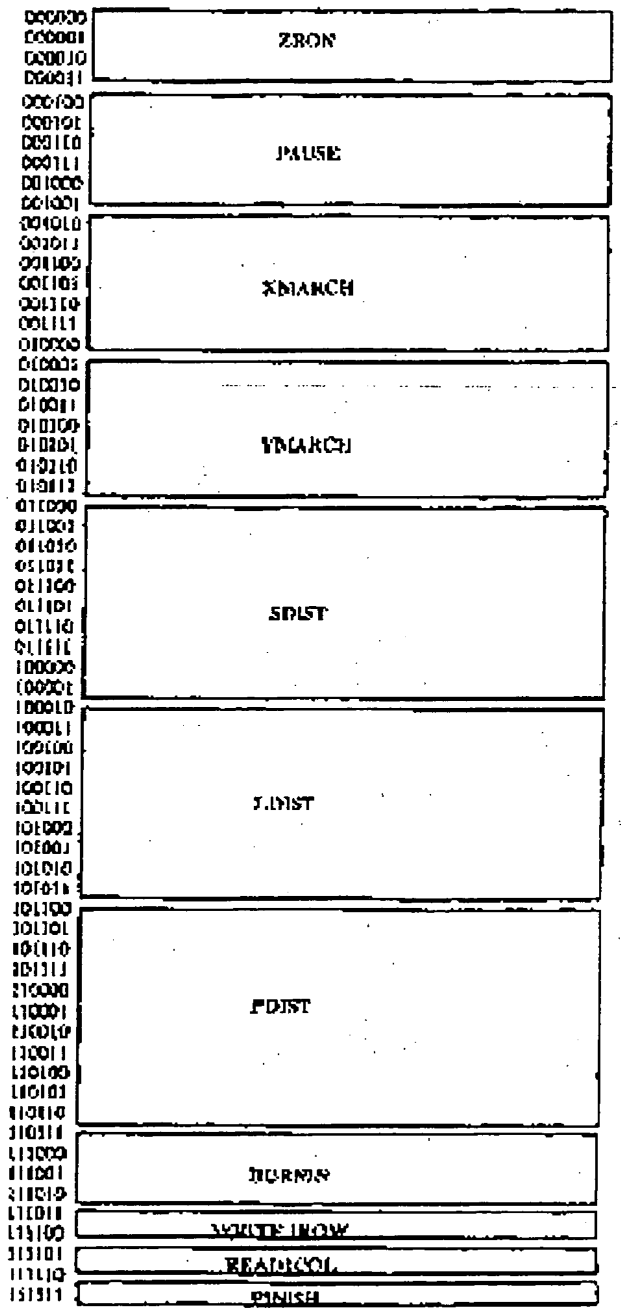
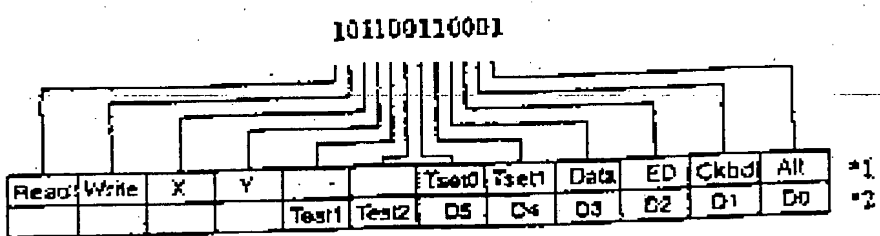


FIG 28



Test1	Test2	Comments
0	0	Test if the test is enabled in enabled tests register 330.
0	1	Test for inverted pattern
1	0	Increment address by 1, and check for overflow (array address)
1	1	Increment counter and check if required value is reached (timing counter)

Tset0	Tset1	Comments
0	0	Timing set A
0	1	Timing set C
1	0	Timing set B
1	1	Timing set A, decrement

*1 Memory Access Instruction

*2 Program Control Instruction

FIG. 29

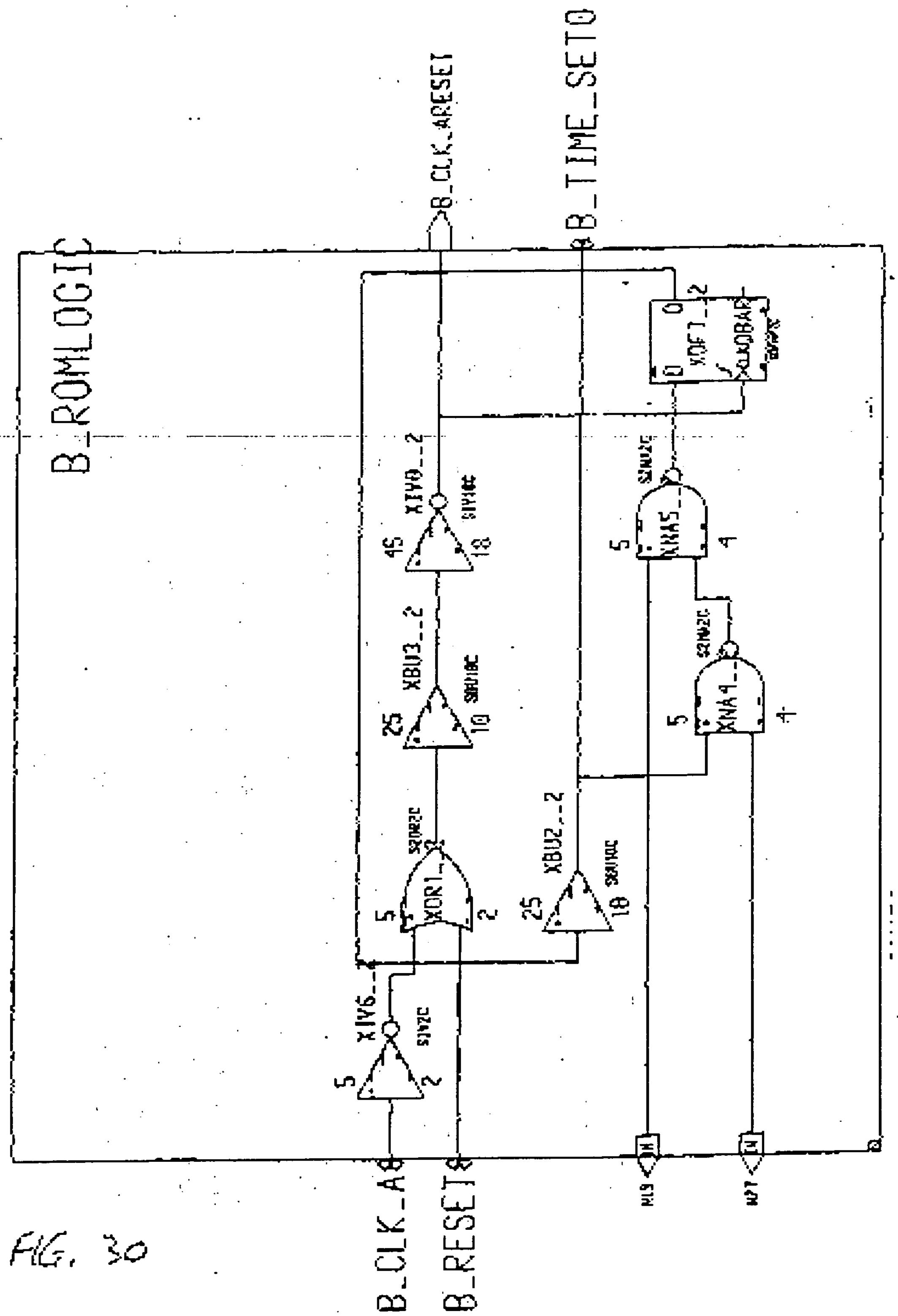

B_ROMLOGIC
B_CLK_ARESET
B_TIME_SET0
B_CLK_A
B_RESET
XIV6
XOR1
XBU2
XBU3
XIV0
XNA5
XNA4
XDF1

FIG. 30

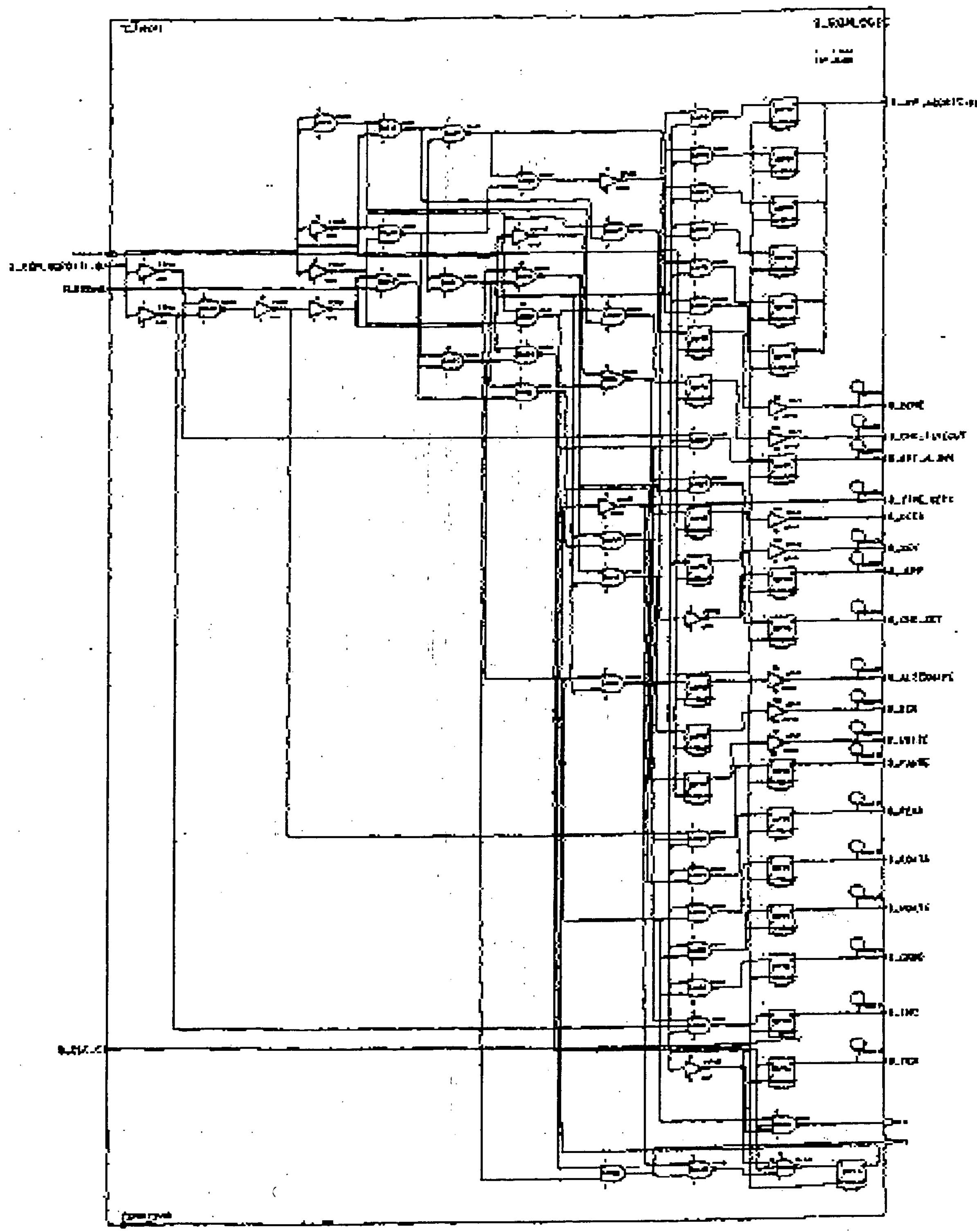


FIG. 31

```
architecture behav of b_passfail is
signal B_EXPECT_DATA   : std_logic_vector (3 downto 0 );
signal B_EXPECT_DATAB  : std_logic_vector (3 downto 0 );
signal B_WRITE_DATA    : std_logic_vector (3 downto 0 );
signal CKBDI,TRUEWRITE,B_REVNVAL,B_RODDVAL,B_WEVNVAL,B_WODDVAL : std_logic;

begin

 CKBDI <= B_CKBD and ( B_LSB_RADD xor B_LSB_TADD);

 pass_fail_check : process(B_CLK,B_OEB,B_RESET)

             begin
               if (B_RESET = '1' ) then
                 B_PF_RESULT <= '0';
               elsif (B_CLK'EVENT and B_CLK = '1') then
                  if (B_OEB = '0' and (B_DQ(31 downto 28) /= B_EXPECT_DATA
                      or B_DQ(27 downto 24) /= B_EXPECT_DATAB)) then
                    B_PF_RESULT <= '1';
                  end if;
               end if;
             end process pass_fail_check;

   B_REVNVAL <= B_RDATA xor CKBDI;
   B_RODDVAL <= B_ALTERNATE xor B_RDATA xor CKBDI;
   B_WEVNVAL <= B_WDATA xor CKBDI;
   B_WODDVAL <= B_ALTERNATE xor B_WDATA xor CKBDI;

   B_EXPECT_DATA(0)  <= B_REVNVAL;
   B_EXPECT_DATA(1)  <= B_RODDVAL;
   B_EXPECT_DATA(2)  <= B_REVNVAL;
   B_EXPECT_DATA(3)  <= B_RODDVAL;
   B_EXPECT_DATAB    <= not B_EXPECT_DATA;

   B_WRITE_DATA(0) <= B_WEVNVAL;
   B_WRITE_DATA(1) <= B_WODDVAL;
   B_WRITE_DATA(2) <= B_WEVNVAL;
   B_WRITE_DATA(3) <= B_WODDVAL;

  monitor_data_values : process(B_WRITE,B_REVNVAL,B_RODDVAL,B_WEVNVAL,B_WODDVAL)
     begin
       if ( B_WRITE = '1') then
         B_DEVEN  <= B_WEVNVAL;
         B_DODD   <= B_WODDVAL;
       else
         B_DEVEN  <= B_REVNVAL;
         B_DODD   <= B_RODDVAL;
      end if;
     end process monitor_data_values;

   TRUEWRITE <= B_WRITE and (not B_WDATA_TIME) and (not TLBRON) and
                (not TLBROWR);

   guarded_block :  block ( TRUEWRITE ='1' )
   begin
       B_DQ(31 downto 28) <= guarded B_WRITE_DATA;
       B_DQ(27 downto 24) <= guarded B_WRITE_DATA;
   end block guarded_block;

end behav;
```

FIG. 32

FIG. 33

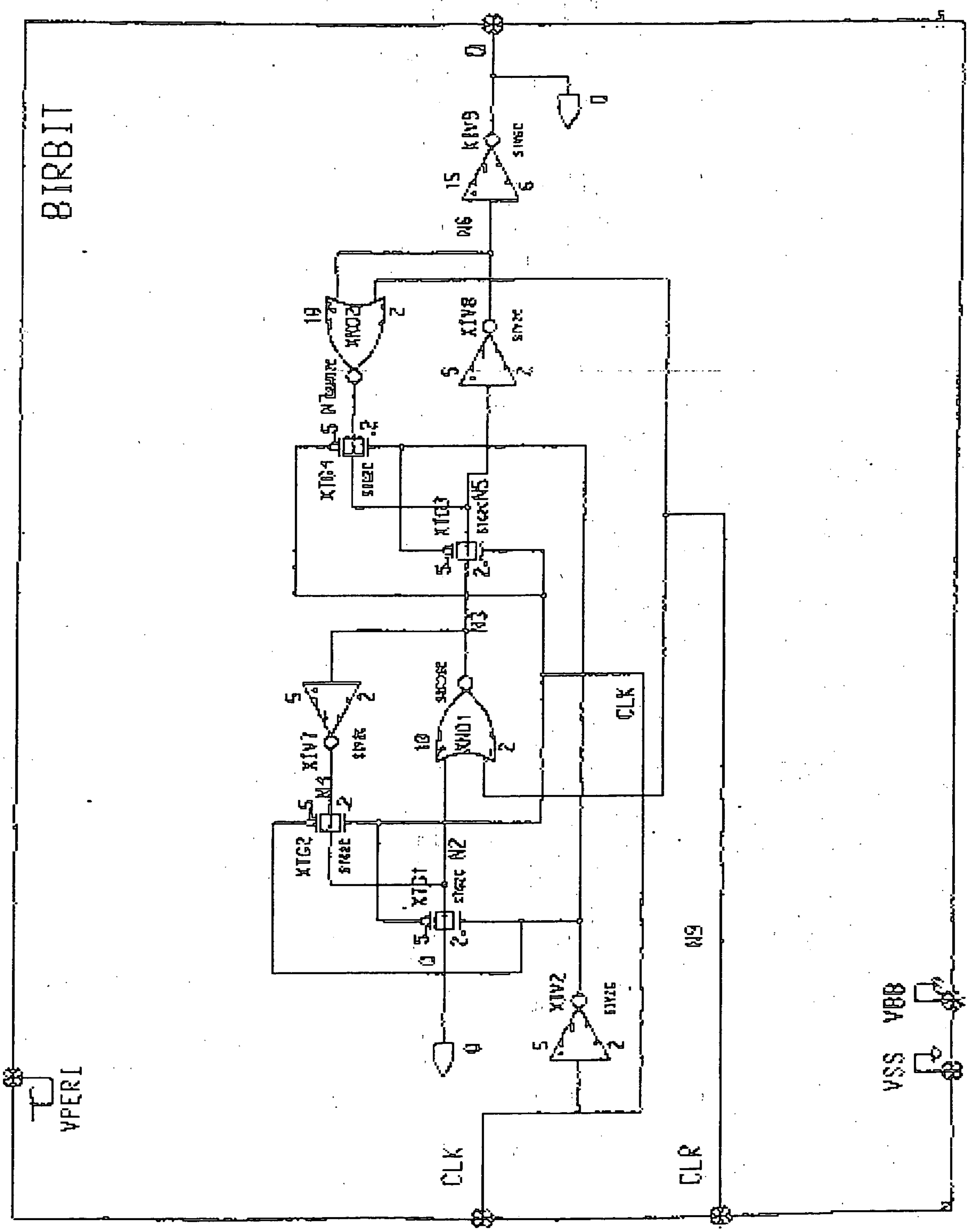
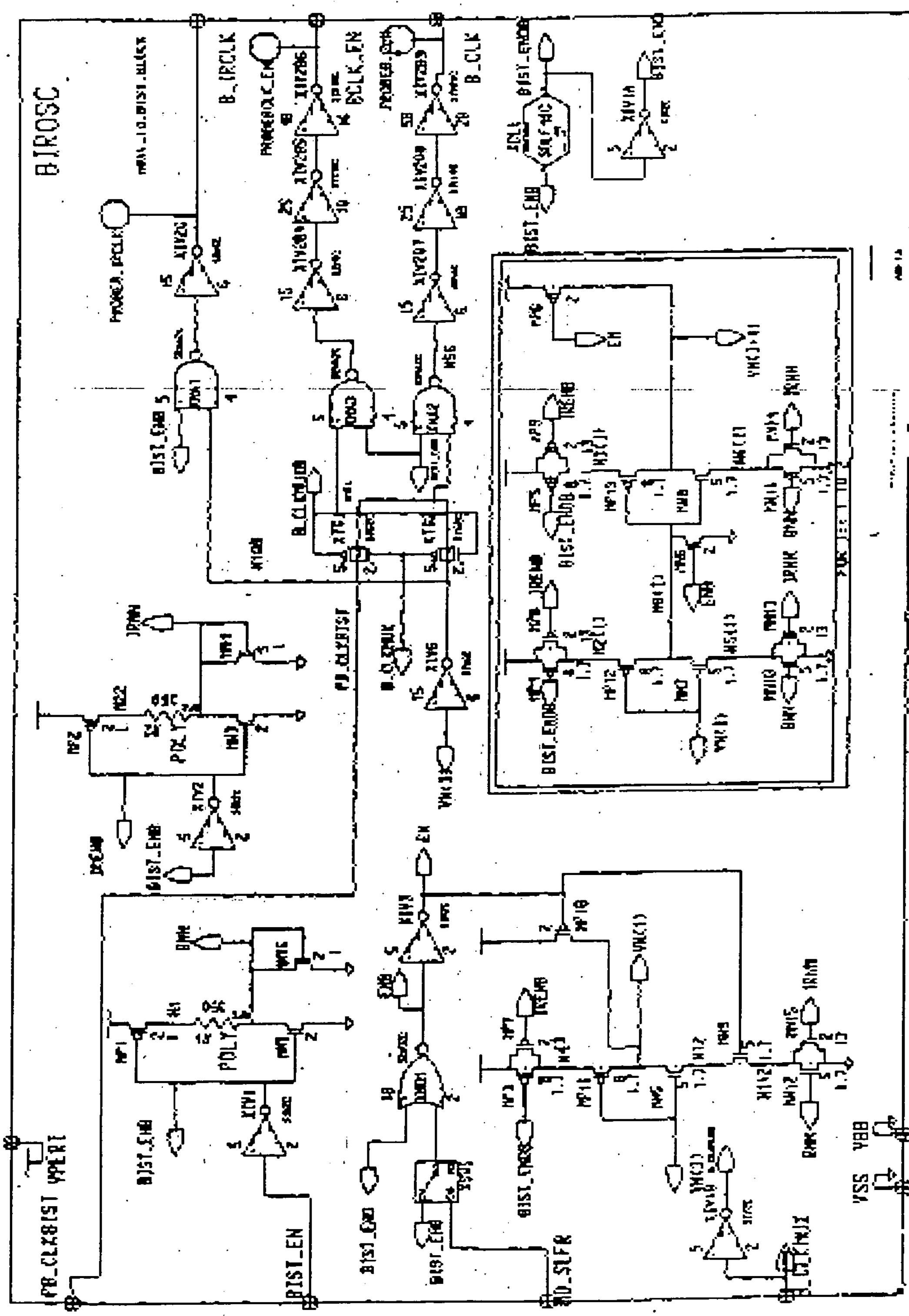

BIRBIT
VPERI
CLK
CLR
VSS
VBB
Q
D
N2
N3
N4
N5
N6
N7
N9
XTG1
XTG2
XTG3
XTG4
XIV2
XIV7
XIV8
XIV9
XNO1
XNO2

FIG. 34



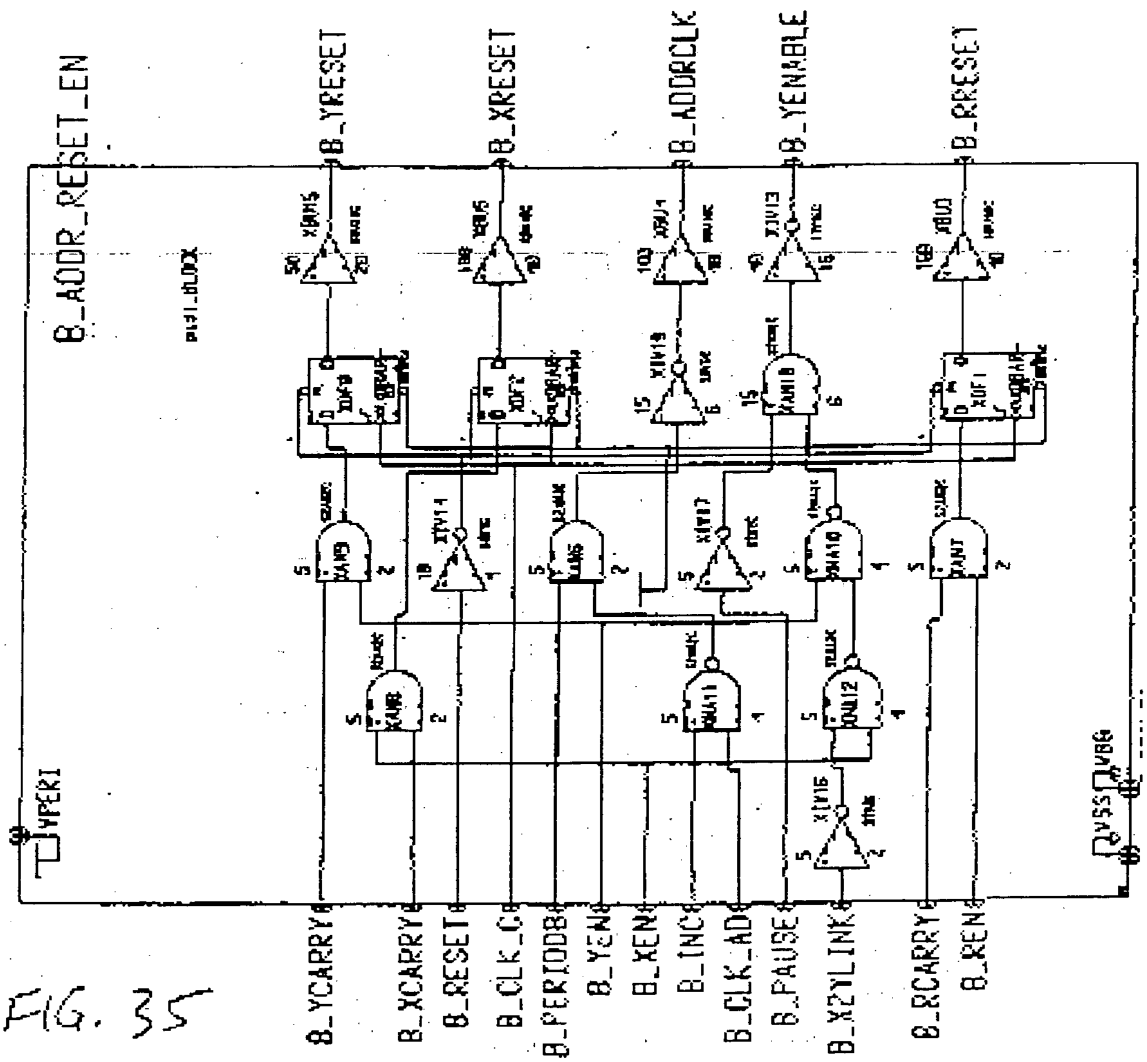

B_ADDR_RESET_EN
B_YRESET
B_XRESET
B_ADDRCLK
B_YENABLE
B_RRESET
VPERI
VSS
VBG
B_YCARRY
B_XCARRY
B_RESET
B_CLK_C
B_PERIODB
B_YEN
B_XEN
B_INC
B_CLK_AD
B_PAUSE
B_X2YLINK
B_RCARRY
B_REN

FIG. 35

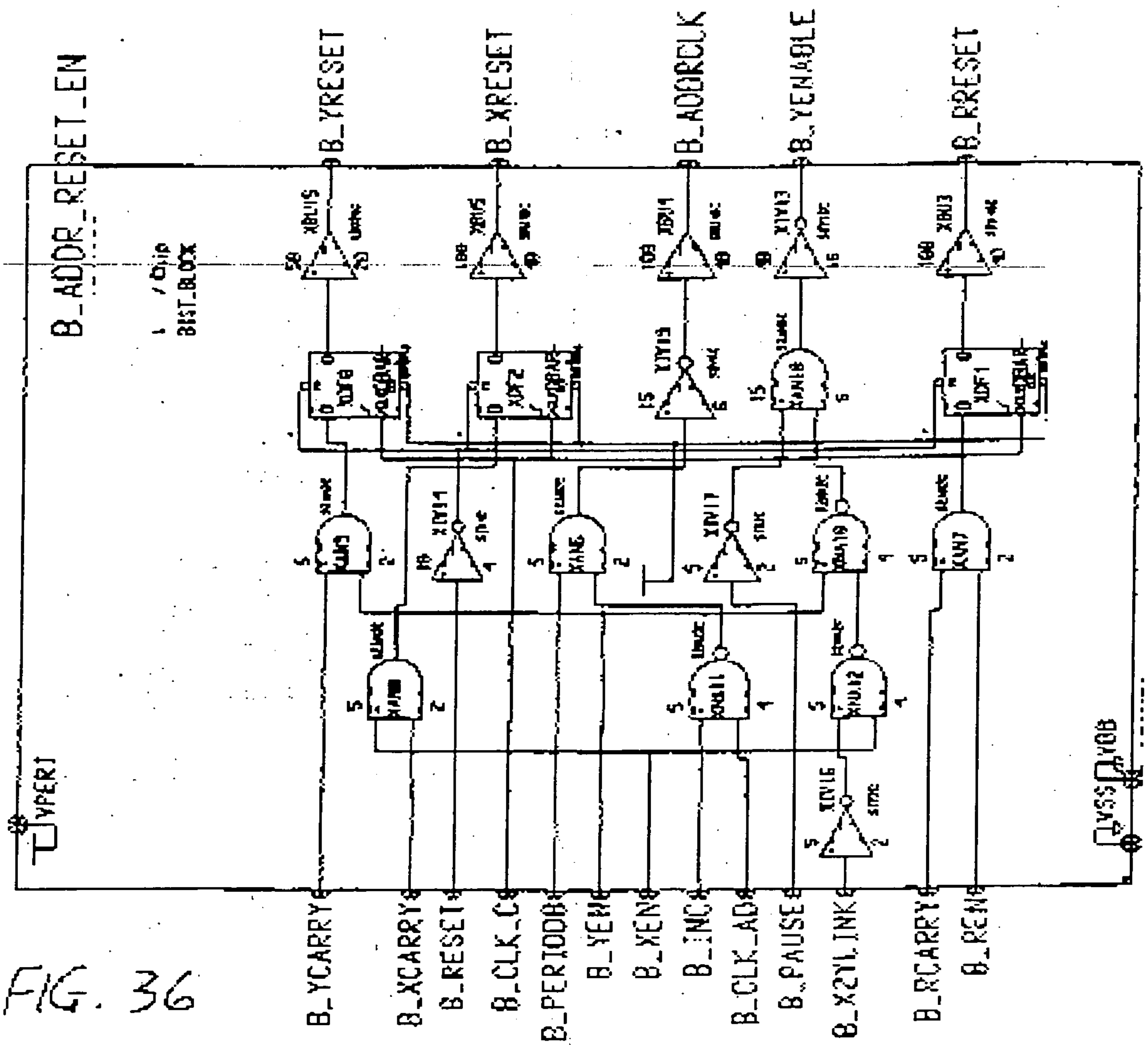

B_ADDR_RESET_EN
BIST_BLOCK
B_YRESET
B_XRESET
B_ADDRCLK
B_YENABLE
B_RRESET
VPERI
VSS
VDD
B_YCARRY
B_XCARRY
B_RESET
B_CLK_C
B_PERIODB
B_YEN
B_XEN
B_INC
B_CLK_AD
B_PAUSE
B_X2YLINK
B_RCARRY
B_REN

FIG. 36

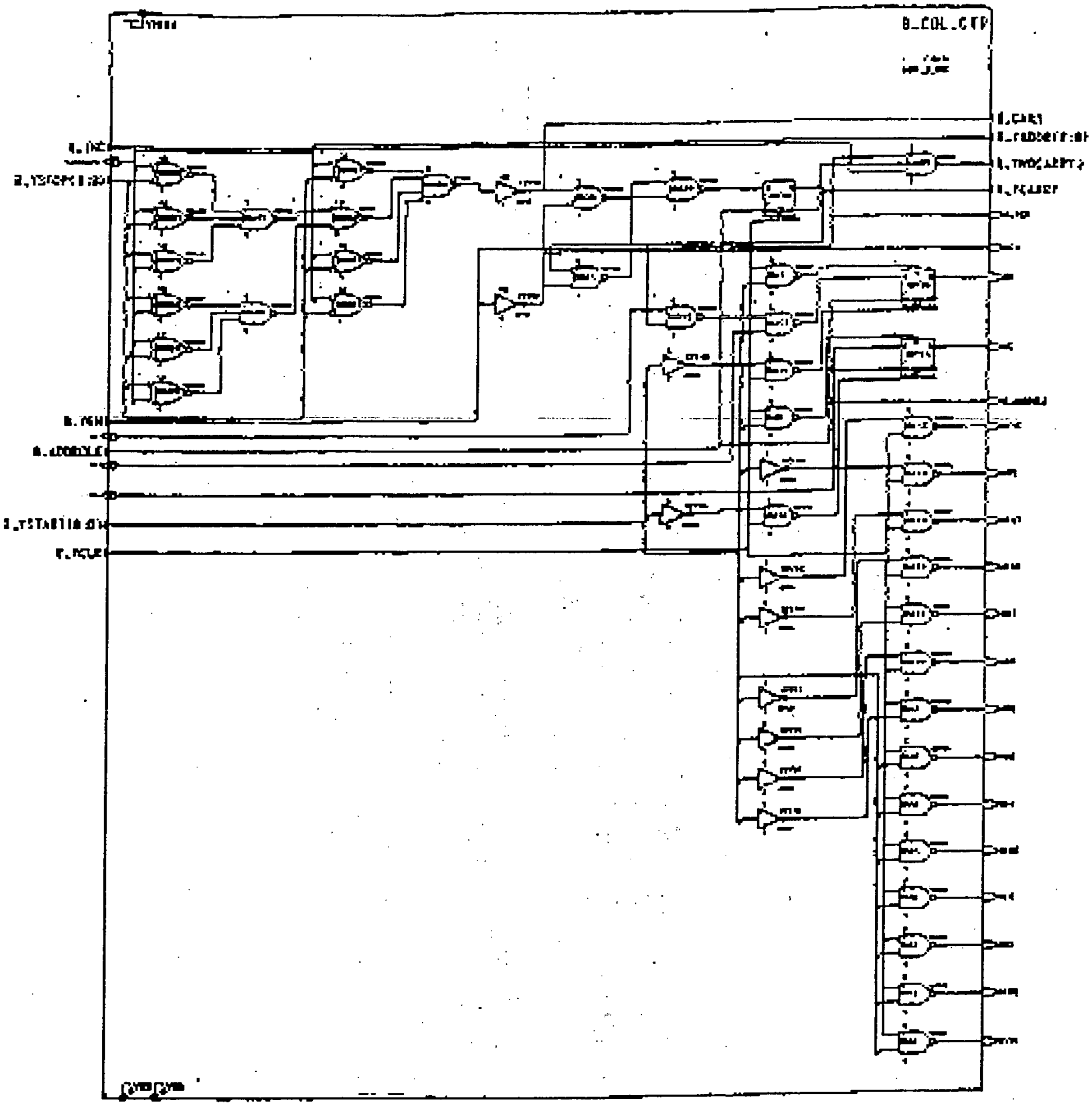


FIG. 37

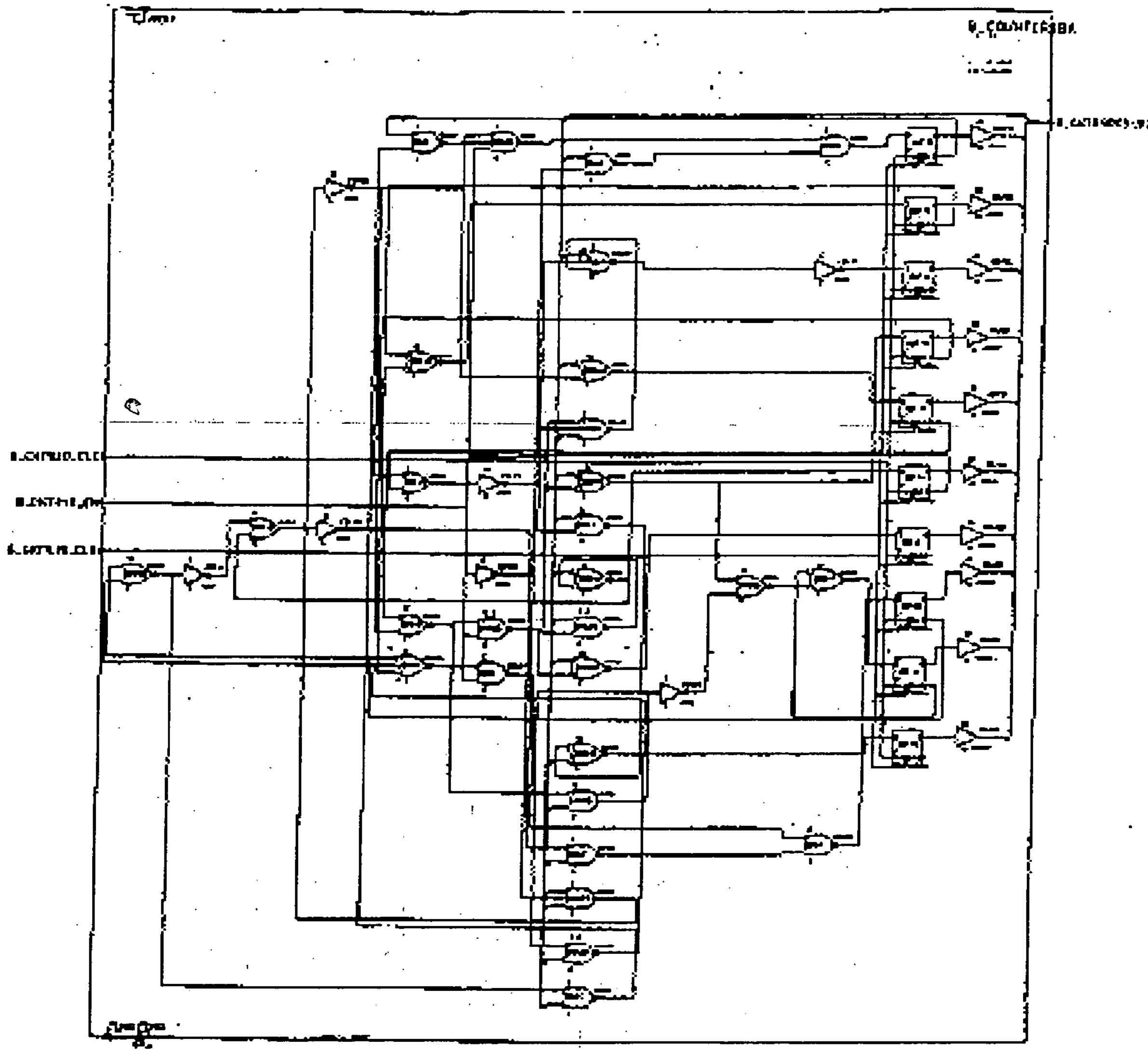


FIG. 38

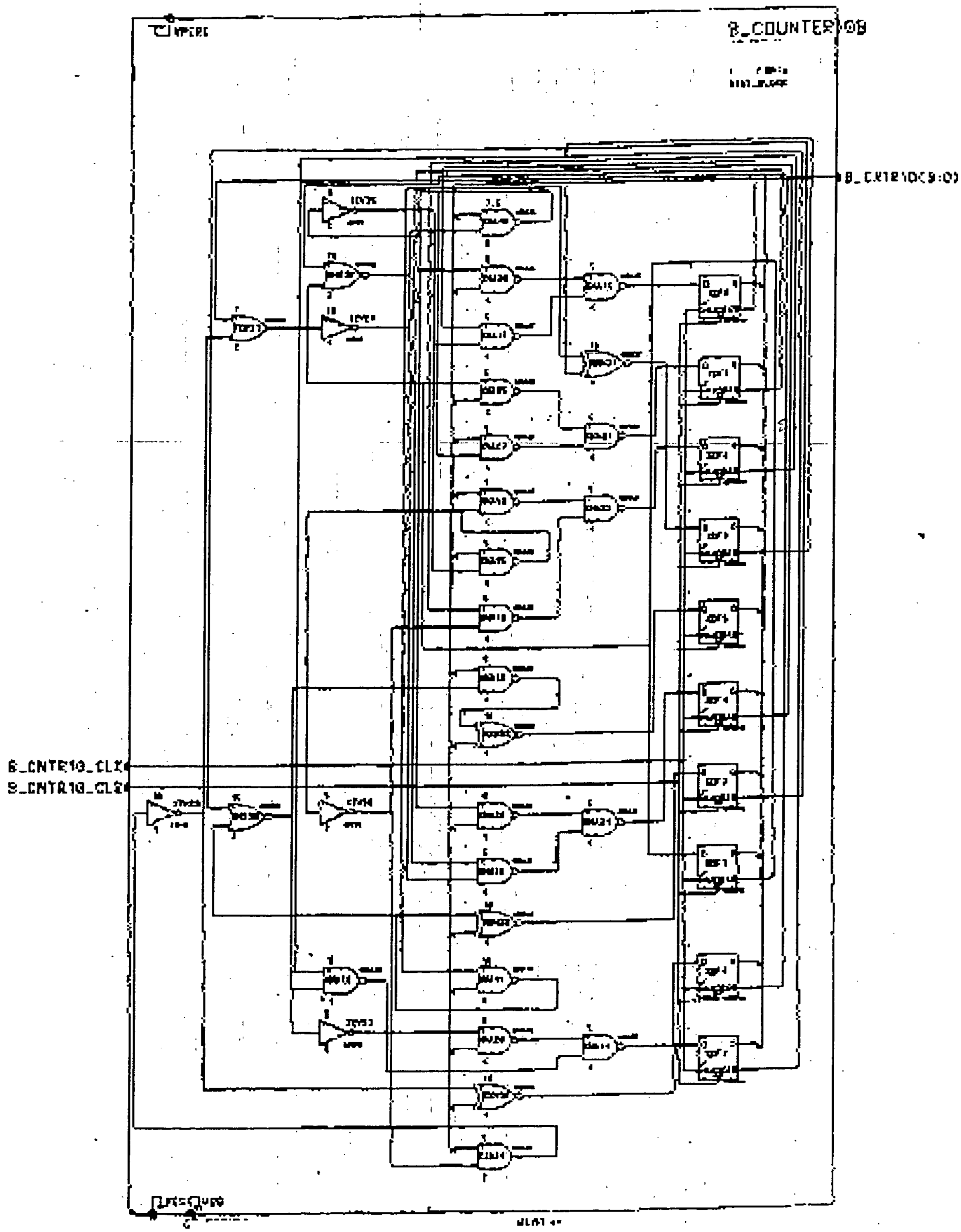


FIG. 39

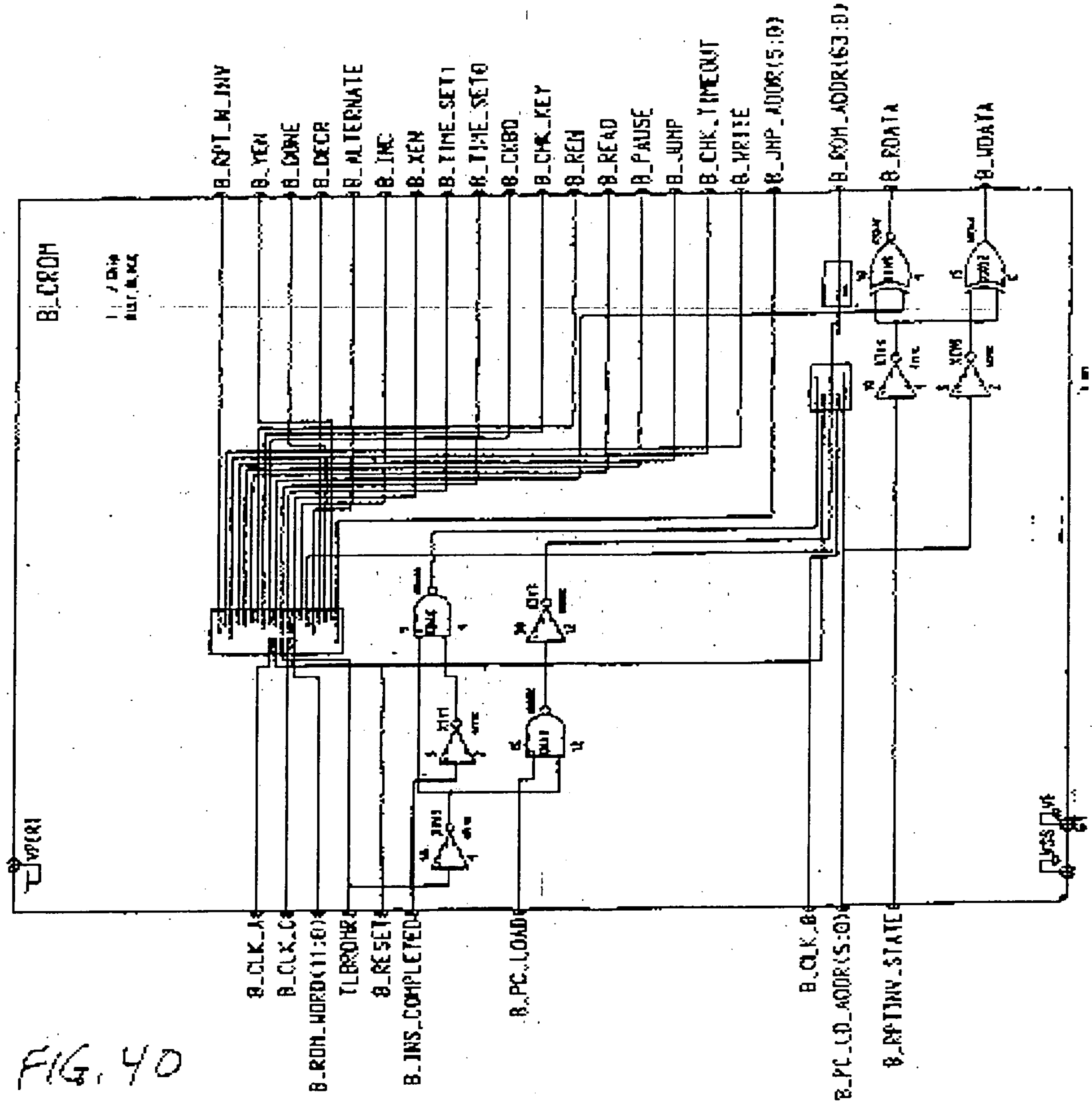

B_CROM
B_RPT_W_INV
B_YEN
B_DONE
B_DECR
B_ALTERNATE
B_INC
B_XEN
B_TIME_SET1
B_TIME_SET0
B_CKBD
B_CHK_KEY
B_REN
B_READ
B_PAUSE
B_JUMP
B_CHK_TIMEOUT
B_WRITE
B_JMP_ADDR(5:0)
B_ROM_ADDR(63:0)
B_RDATA
B_WDATA
B_CLK_A
B_CLK_C
B_ROM_WORD(11:0)
B_RESET
B_INS_COMPLETED
B_PC_LOAD
B_CLK_B
B_PC_LD_ADDR(5:0)
B_RPTINV_STATE

FIG. 40

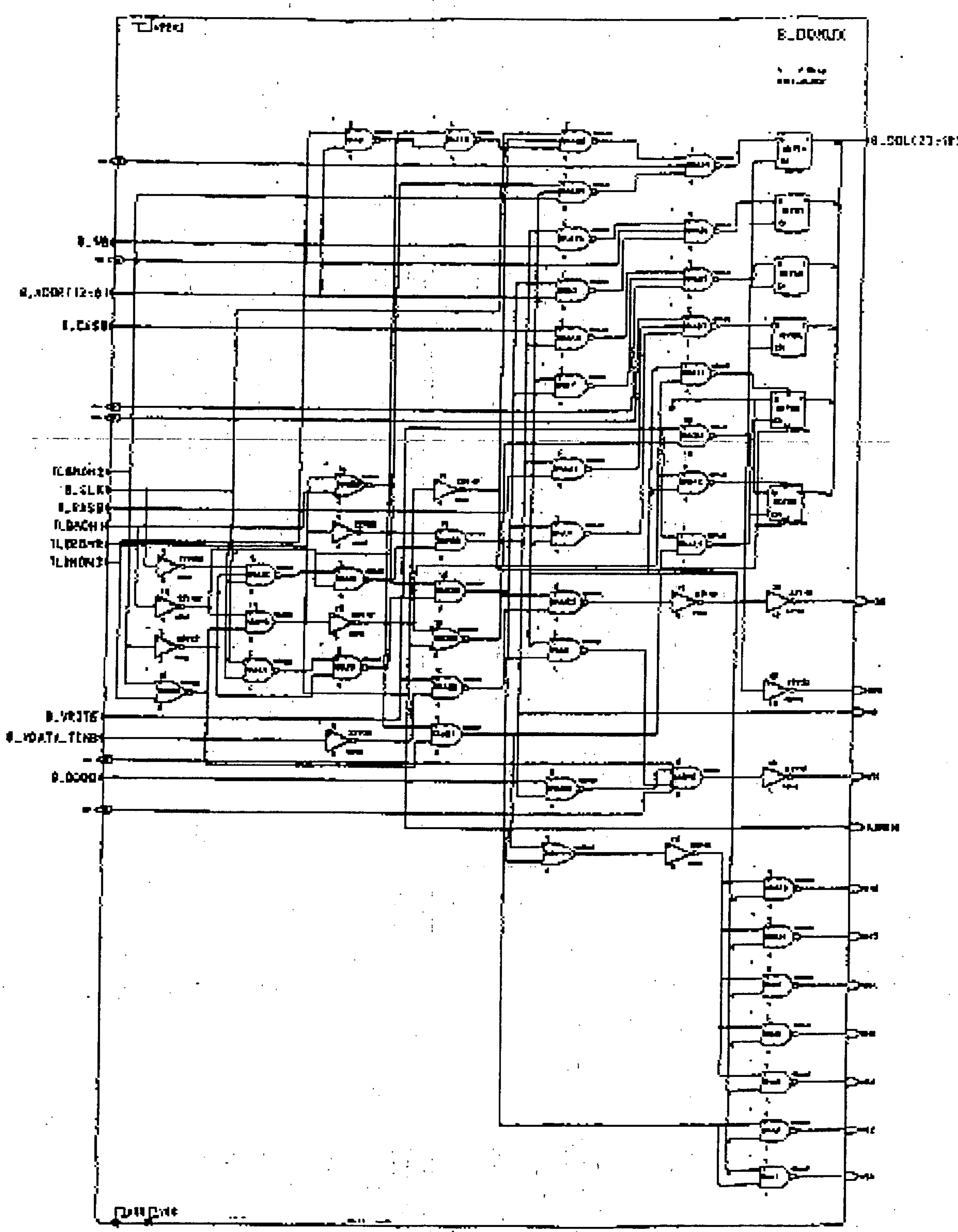


FIG. 41

TI 23011

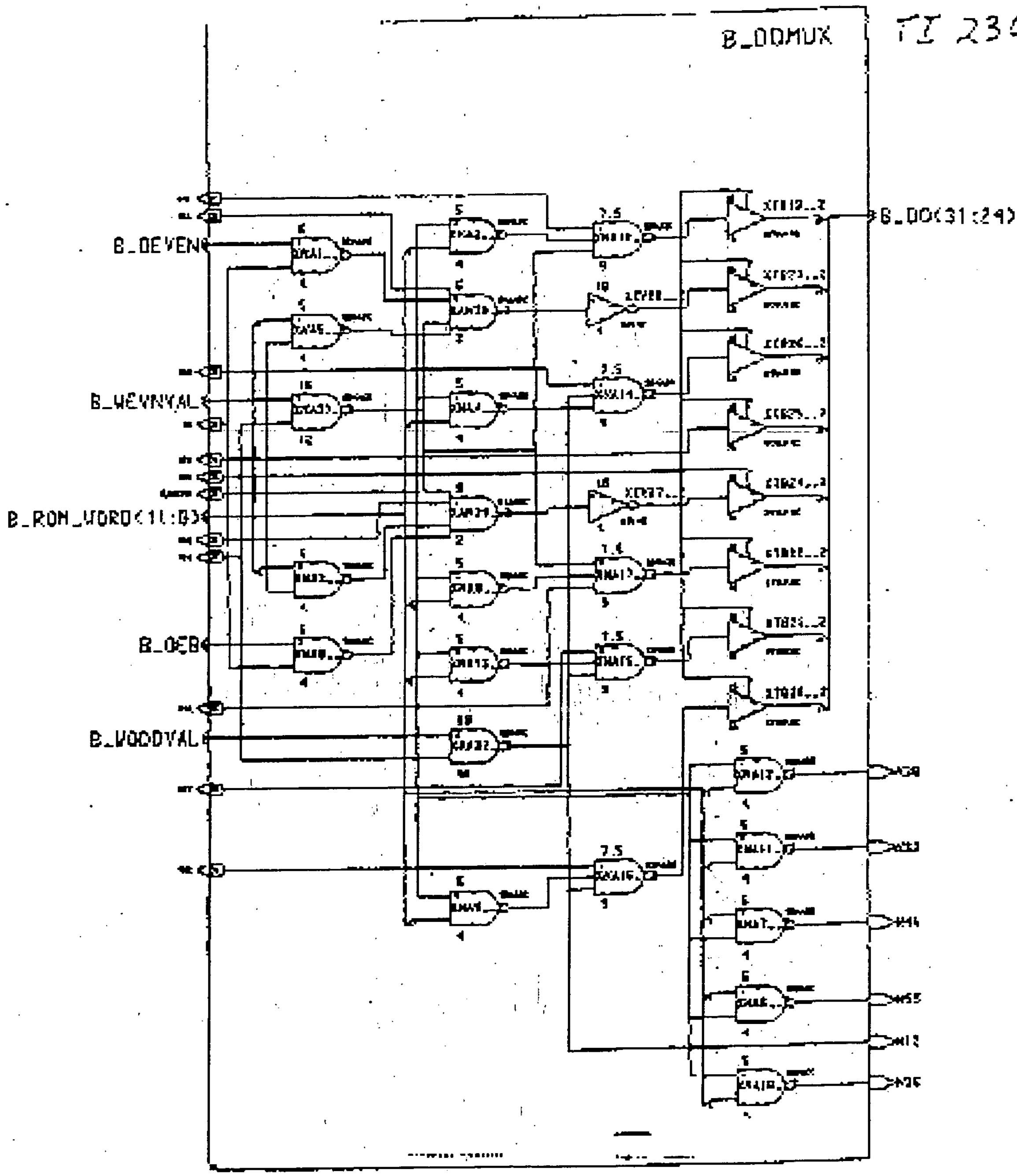


FIG. 42

TI 23 9/

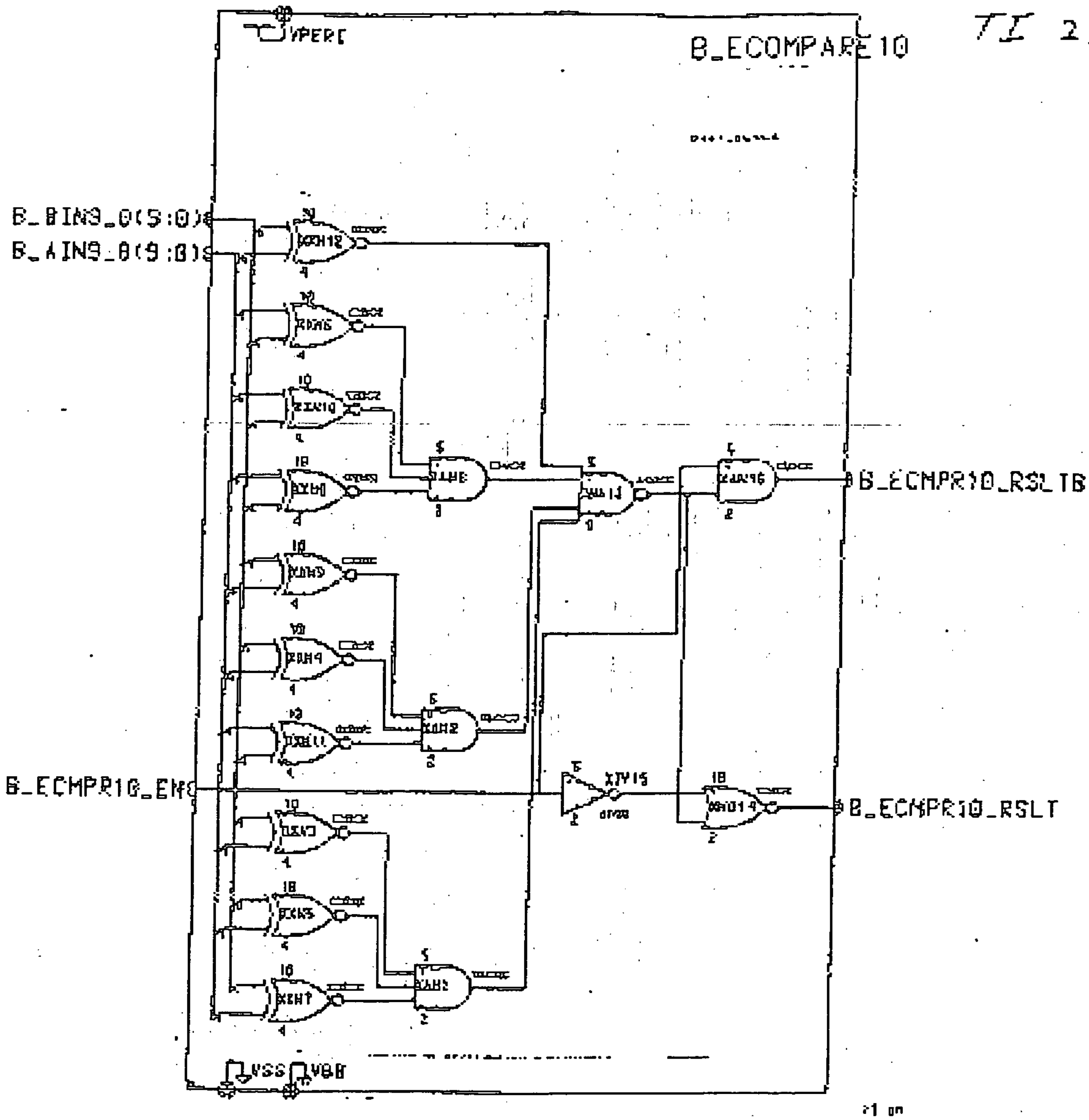

B_ECOMPARE10
B_BIN9_0(9:0)
B_AIN9_0(9:0)
B_ECMPR10_EN
B_ECMPR10_RSLTB
B_ECMPR10_RSLT
VPERI
VSS
VBB

FIG. 43

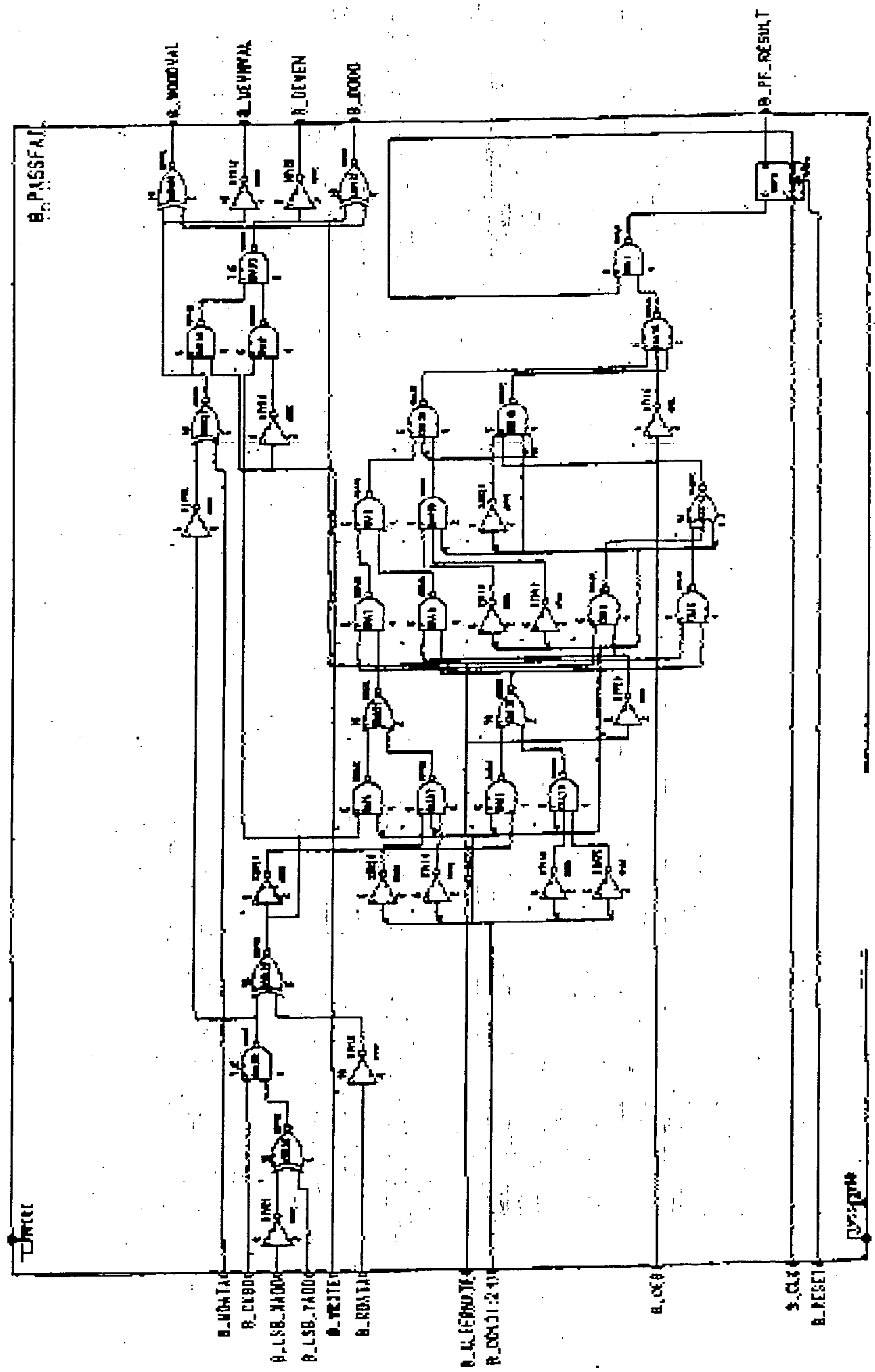

B_PASSFAIL
B_DEVEN
B_DODD
B_PF_RESULT
B_WRITE
B_RDATA
B_CEB
$_CLK
B_RESET

FIG. 44

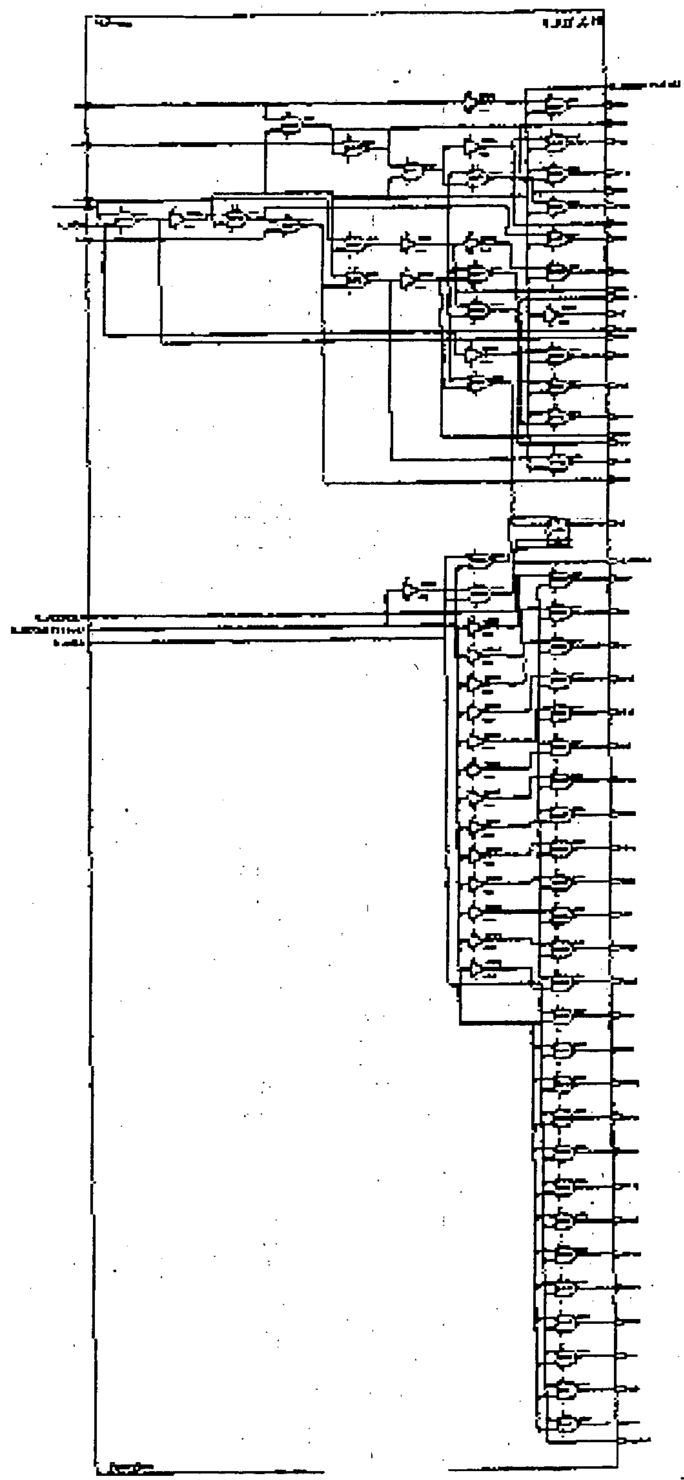


FIG. 45

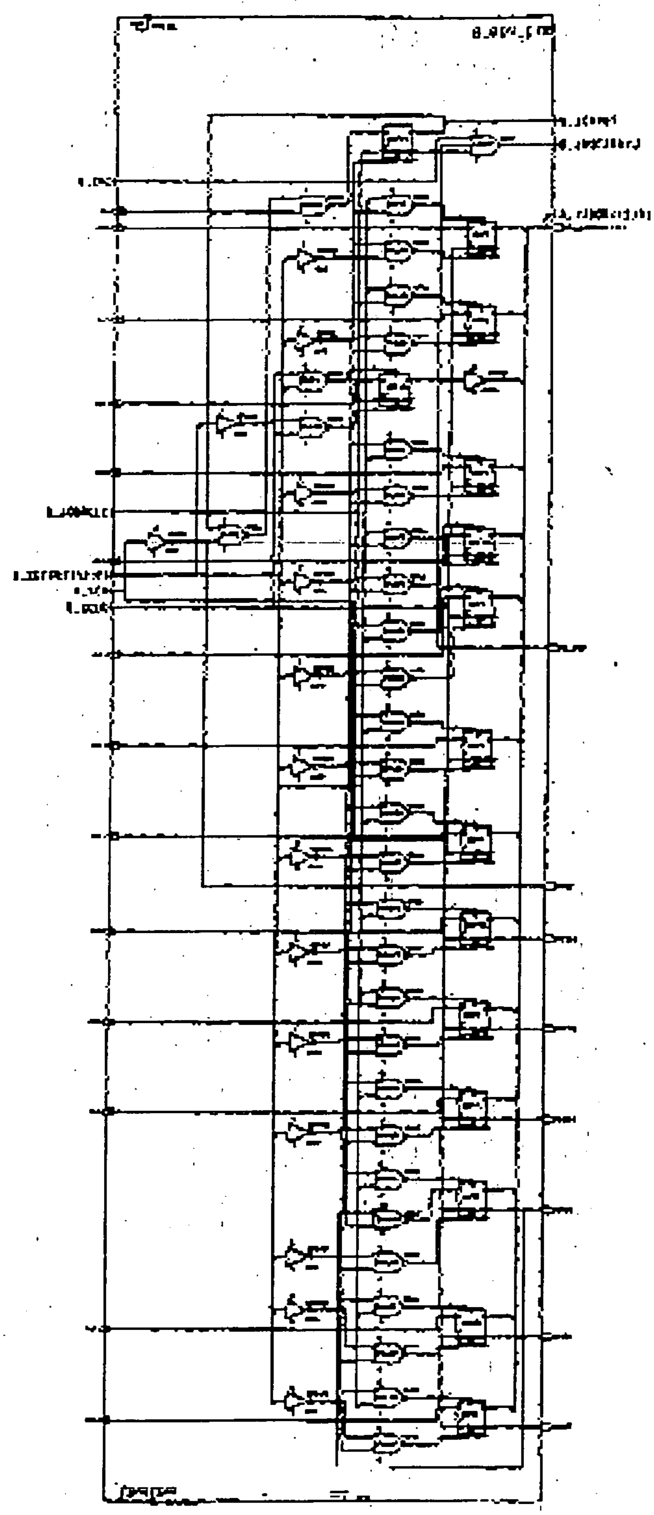


FIG. 46

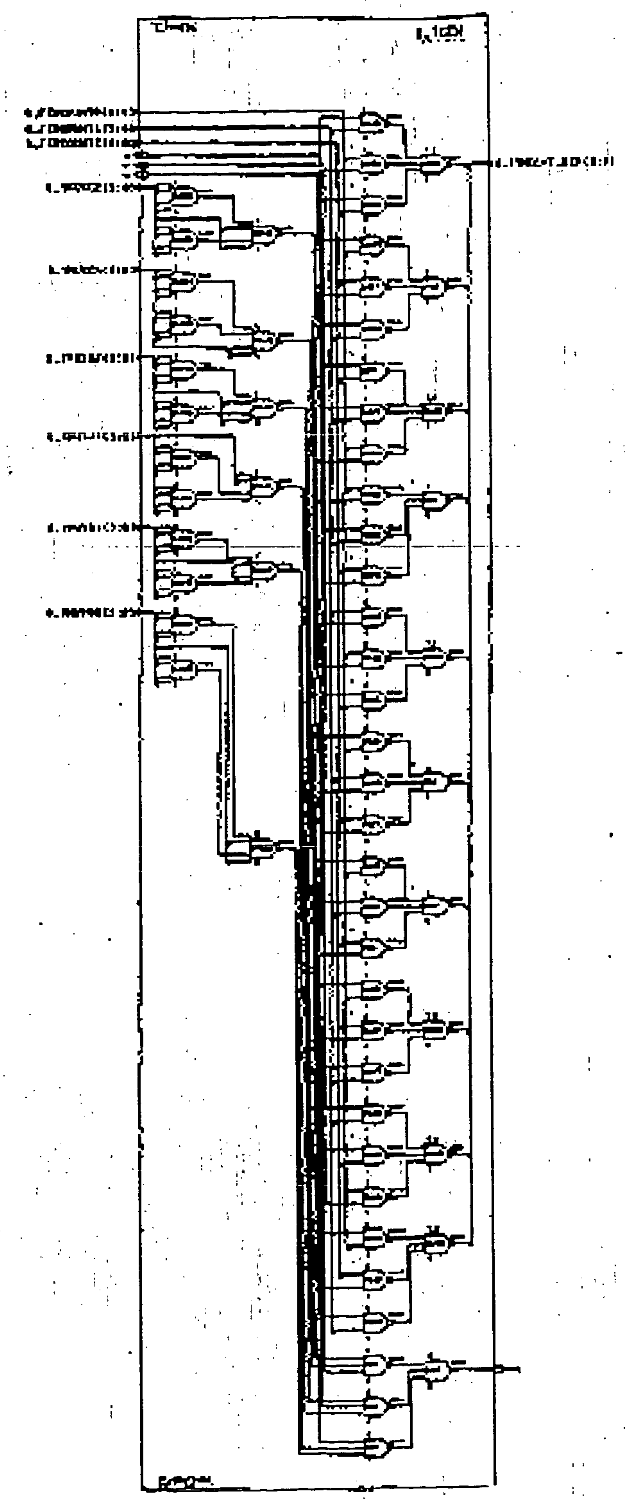


FIG. 47

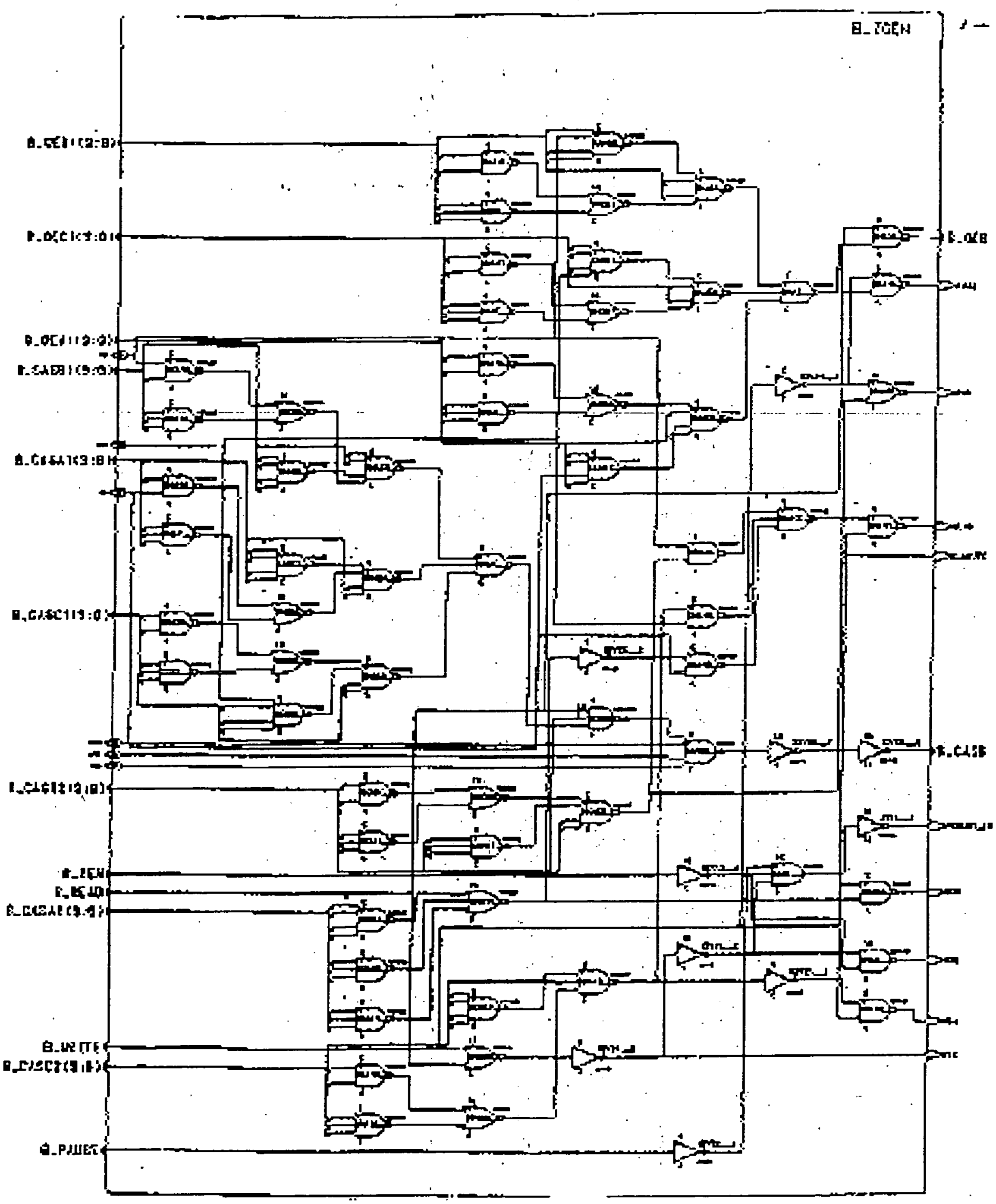


FIG. 48

TC 2

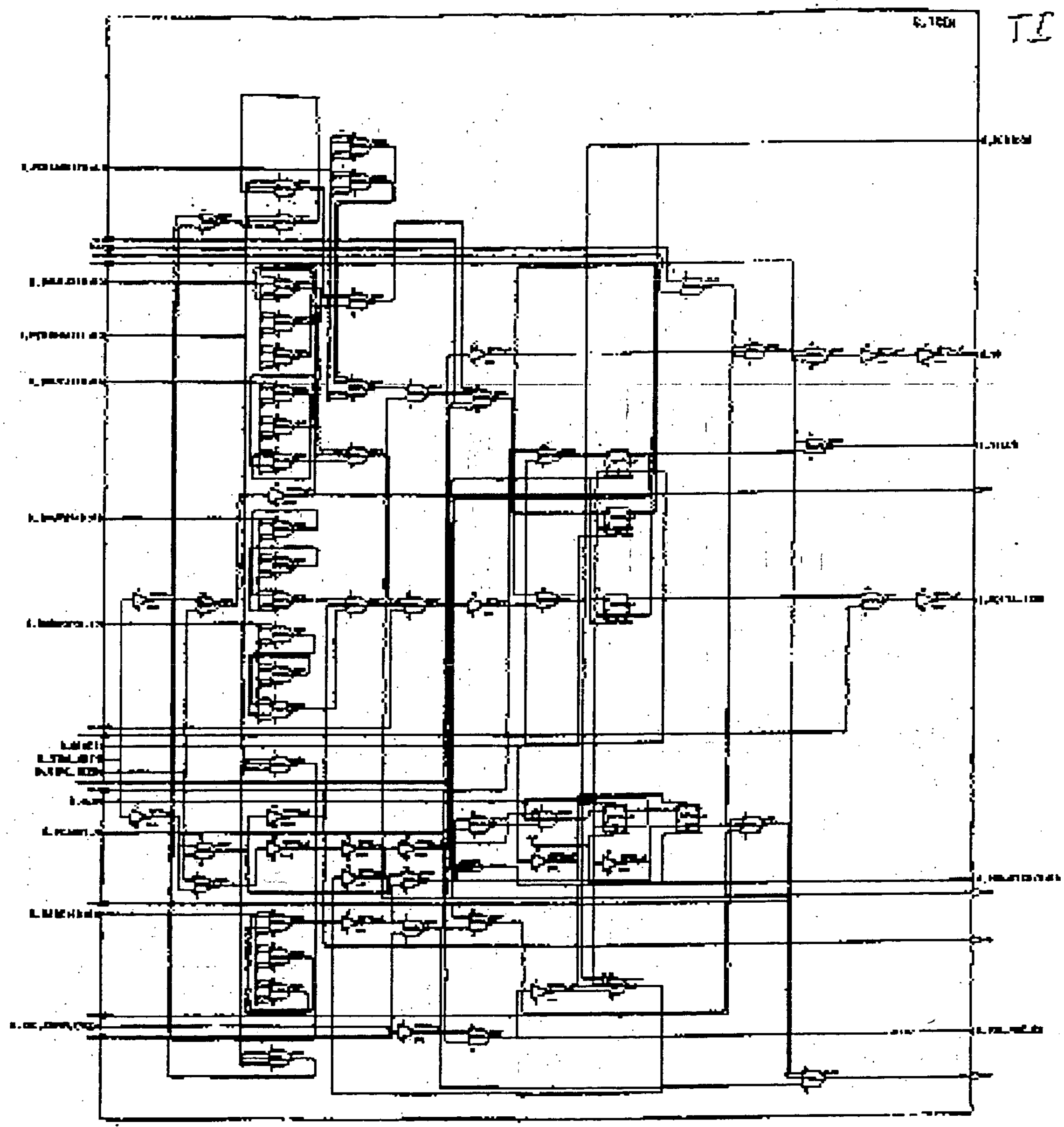


FIG. 49

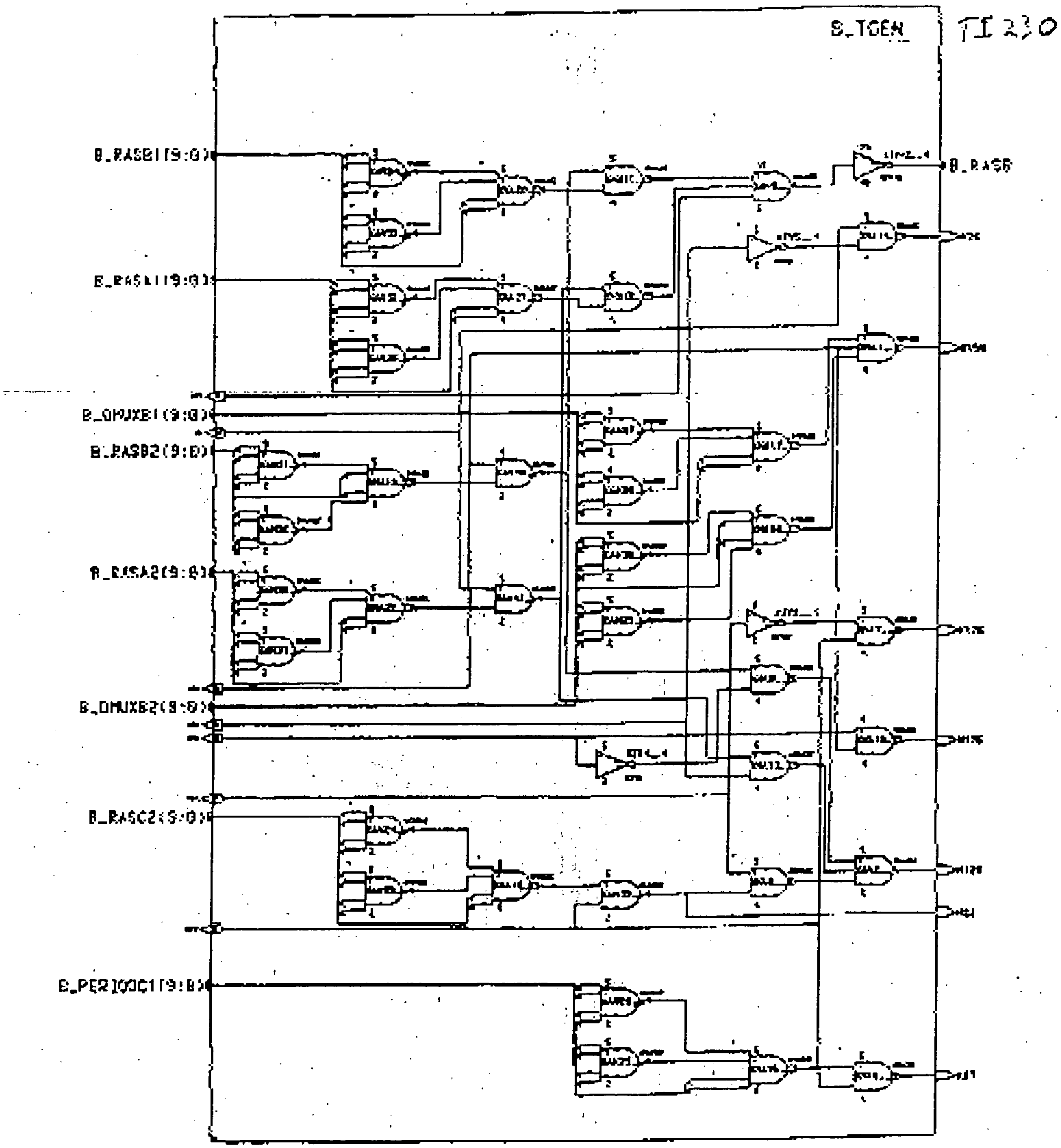

B_TGEN
TI 230
B_RASB1(9:0)
B_RASA1(9:0)
B_DMUXB1(9:0)
B_RASB2(9:0)
B_RASA2(9:0)
B_DMUXB2(9:0)
B_RASC2(9:0)
B_PERIODC1(9:0)
B_RASB

FIG. 50

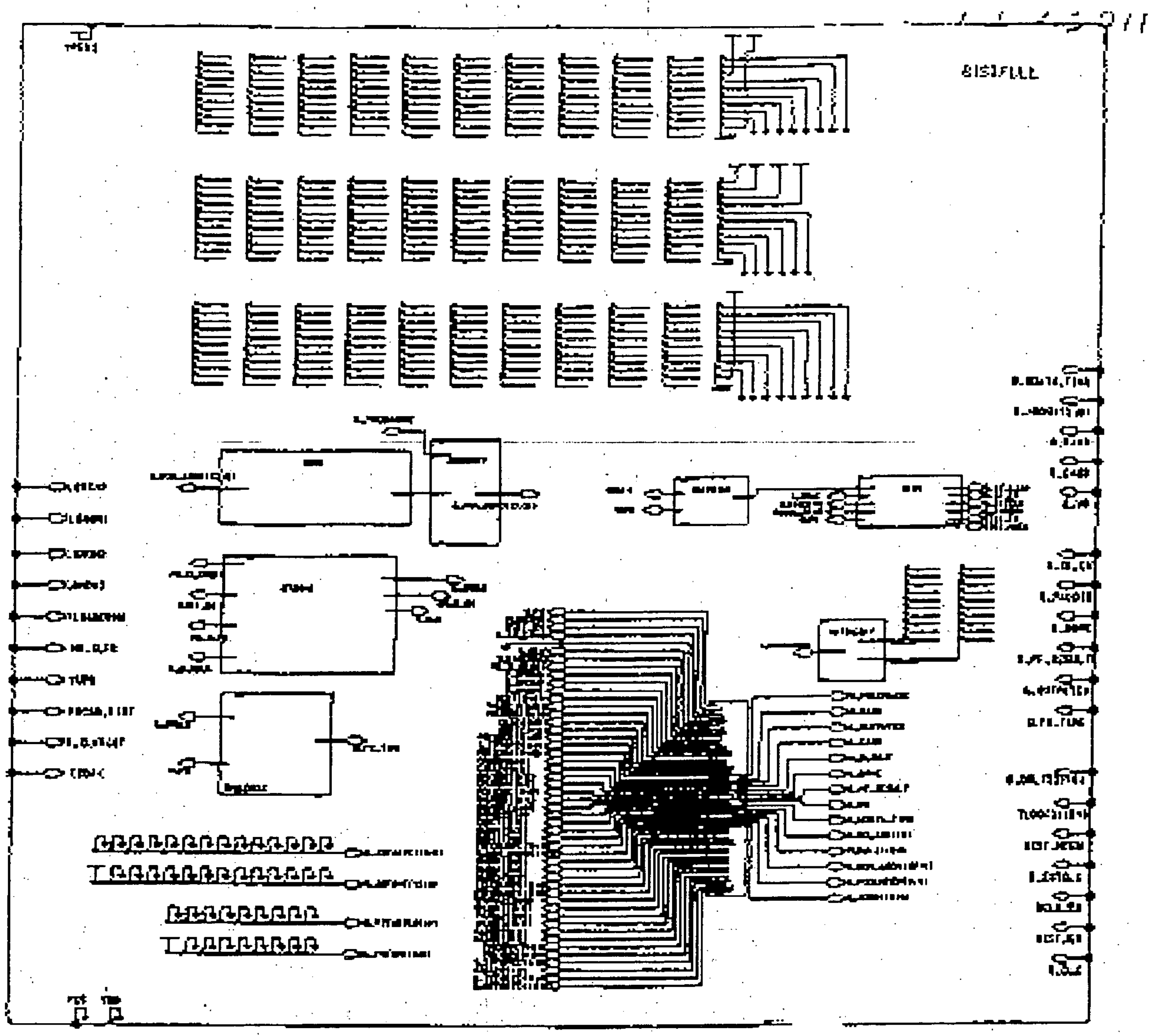


FIG. 51

TI 230

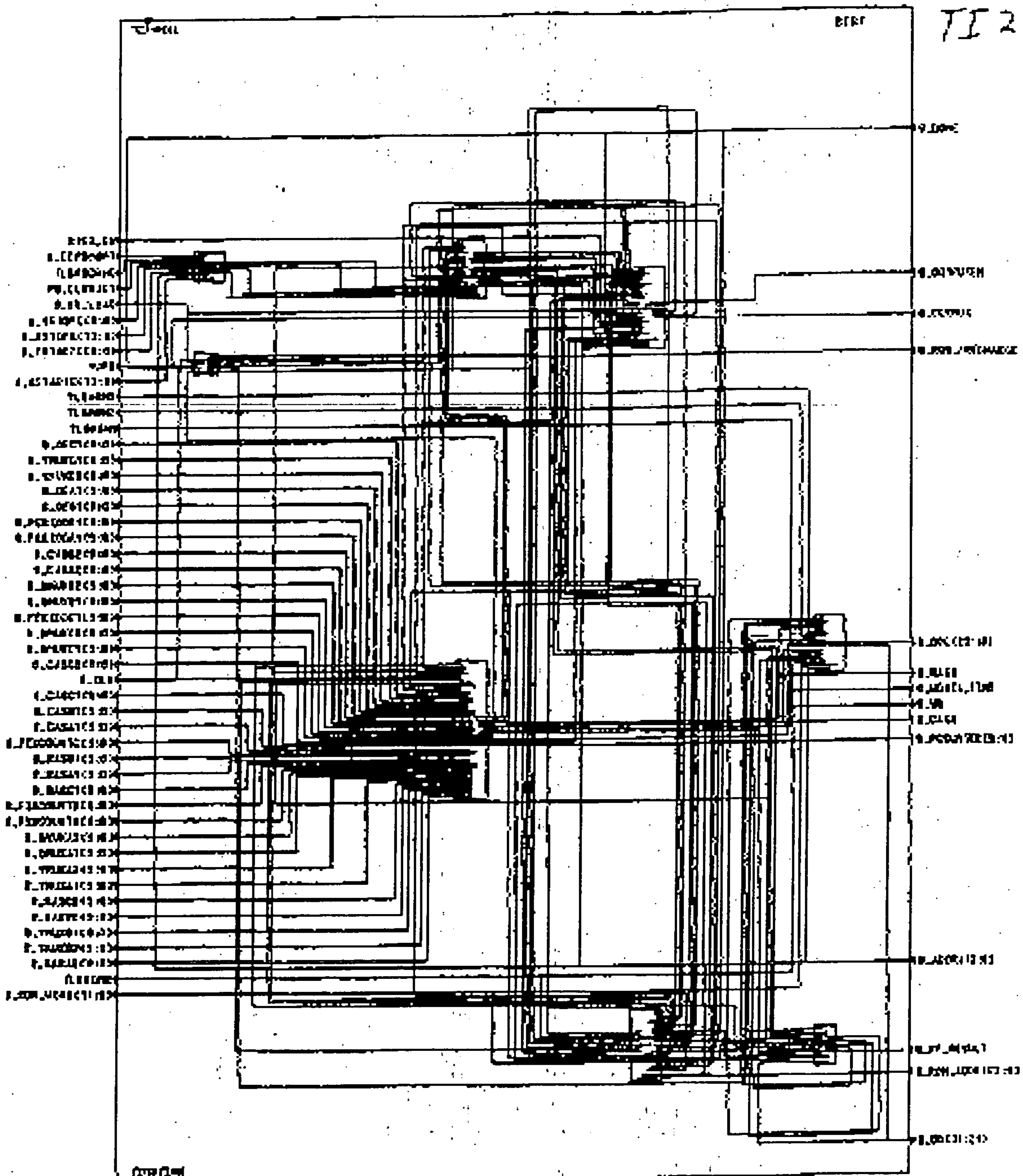


FIG. 52

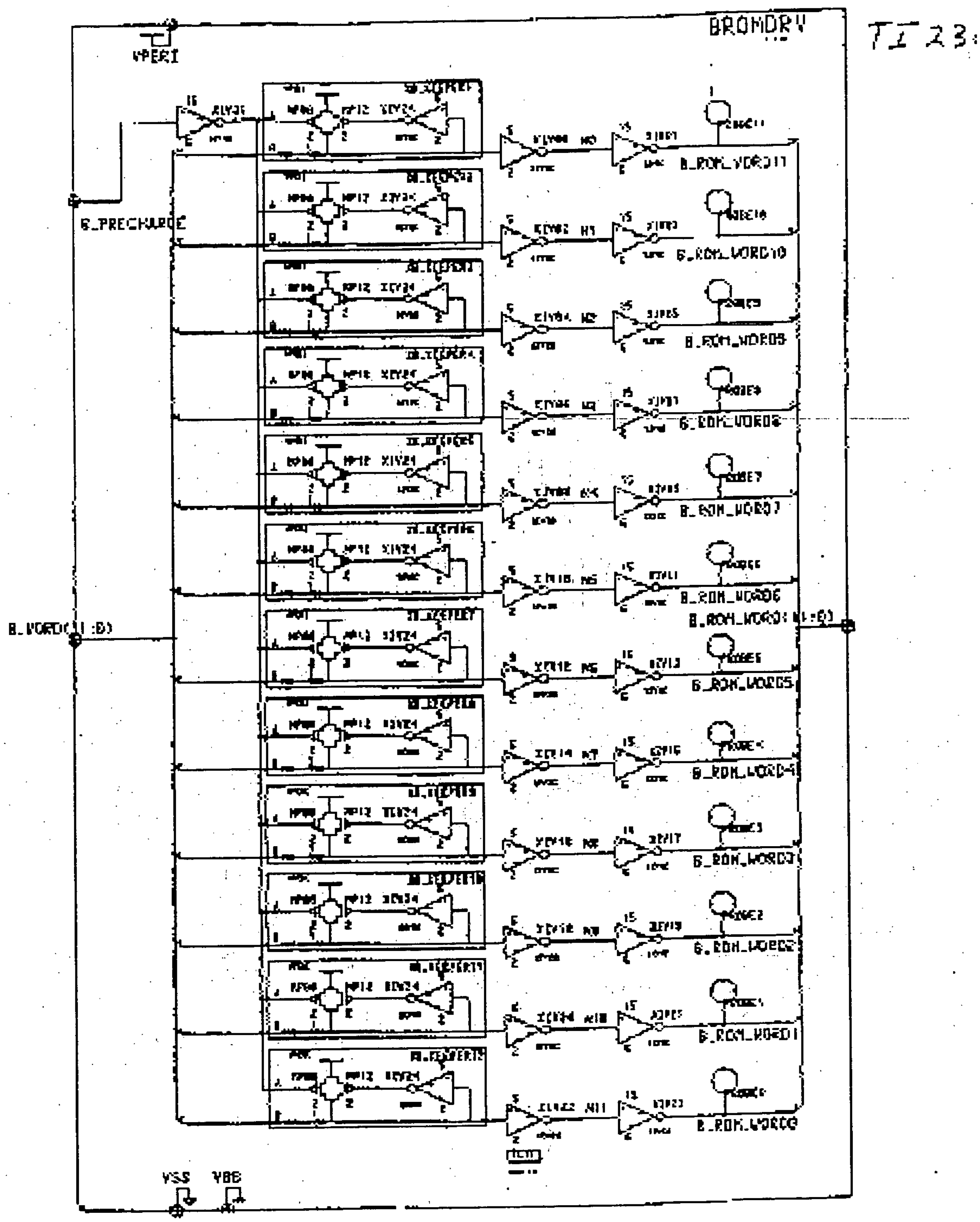

BROMDRV
TI 23:
VPERI
B_PRECHARGE
B_WORD(11:0)
B_ROM_WORD11
B_ROM_WORD10
B_ROM_WORD9
B_ROM_WORD8
B_ROM_WORD7
B_ROM_WORD6
B_ROM_WORD(11:0)
B_ROM_WORD5
B_ROM_WORD4
B_ROM_WORD3
B_ROM_WORD2
B_ROM_WORD1
B_ROM_WORD0
VSS
VBB

FIG. 53

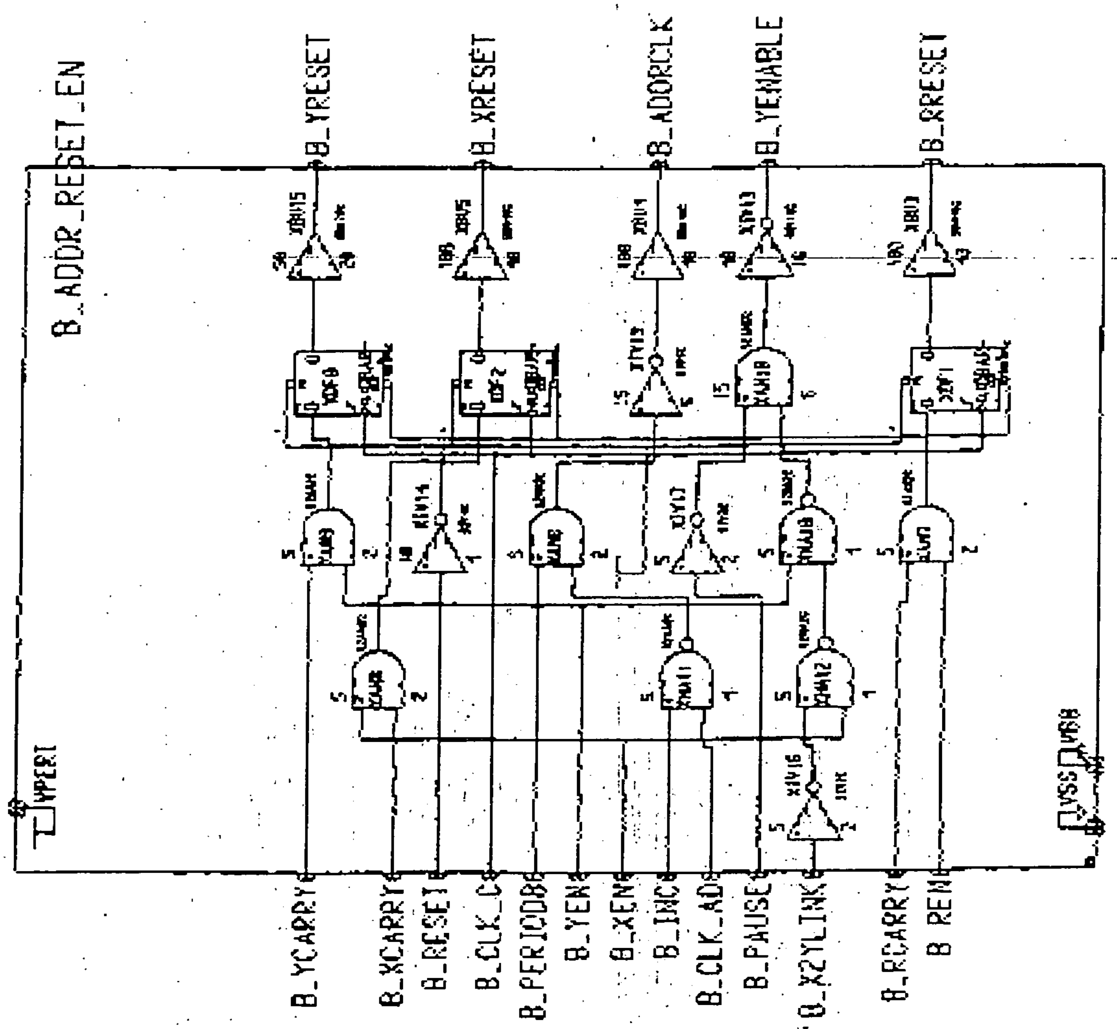

B_ADDR_RESET_EN
B_YRESET
B_XRESET
B_ADDRCLK
B_YENABLE
B_RRESET
VPERI
VSS
VBB
B_YCARRY
B_XCARRY
B_RESET
B_CLK_C
B_PERIODB
B_YEN
B_XEN
B_INC
B_CLK_AD
B_PAUSE
B_X2YLINK
B_RCARRY
B_REN

FIG. 54

TI 23011

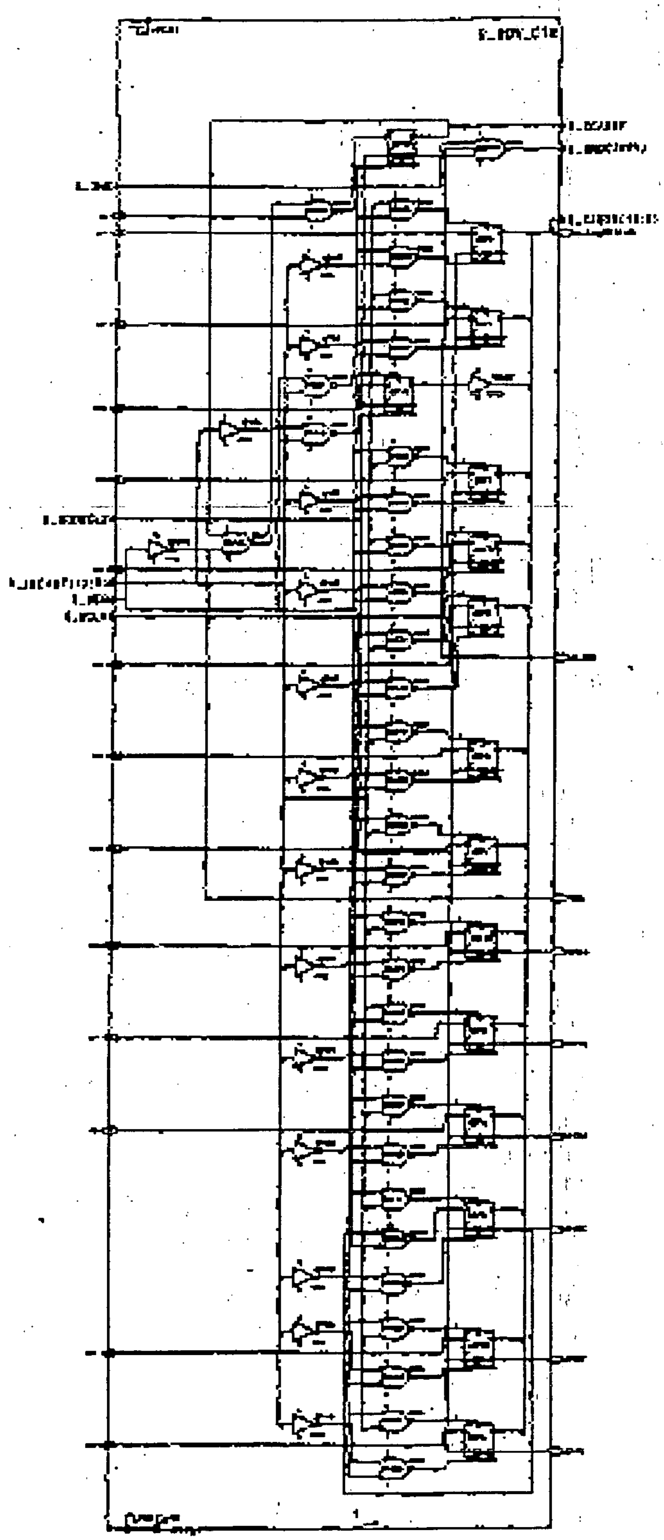


FIG. 55

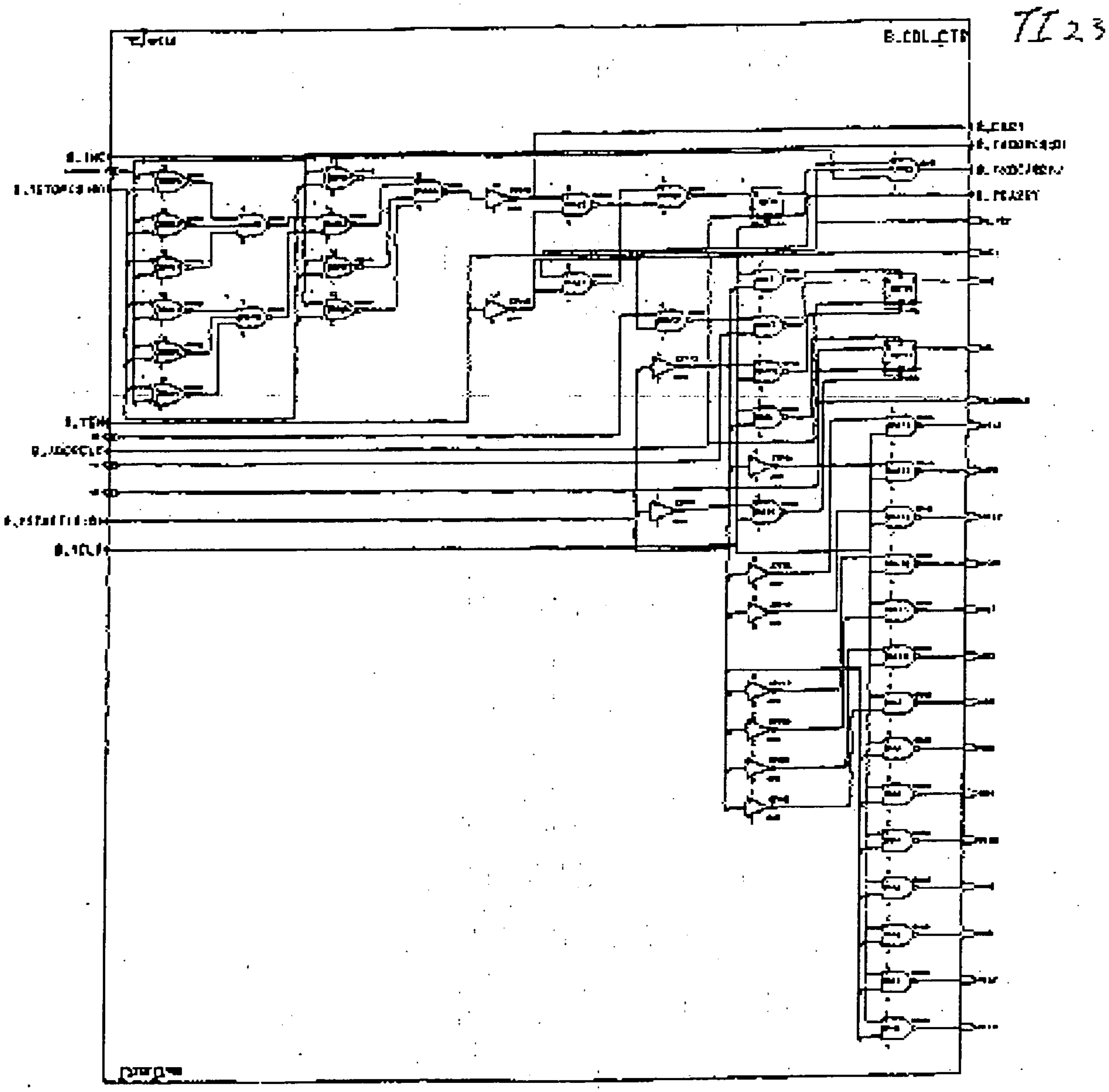


FIG. 56

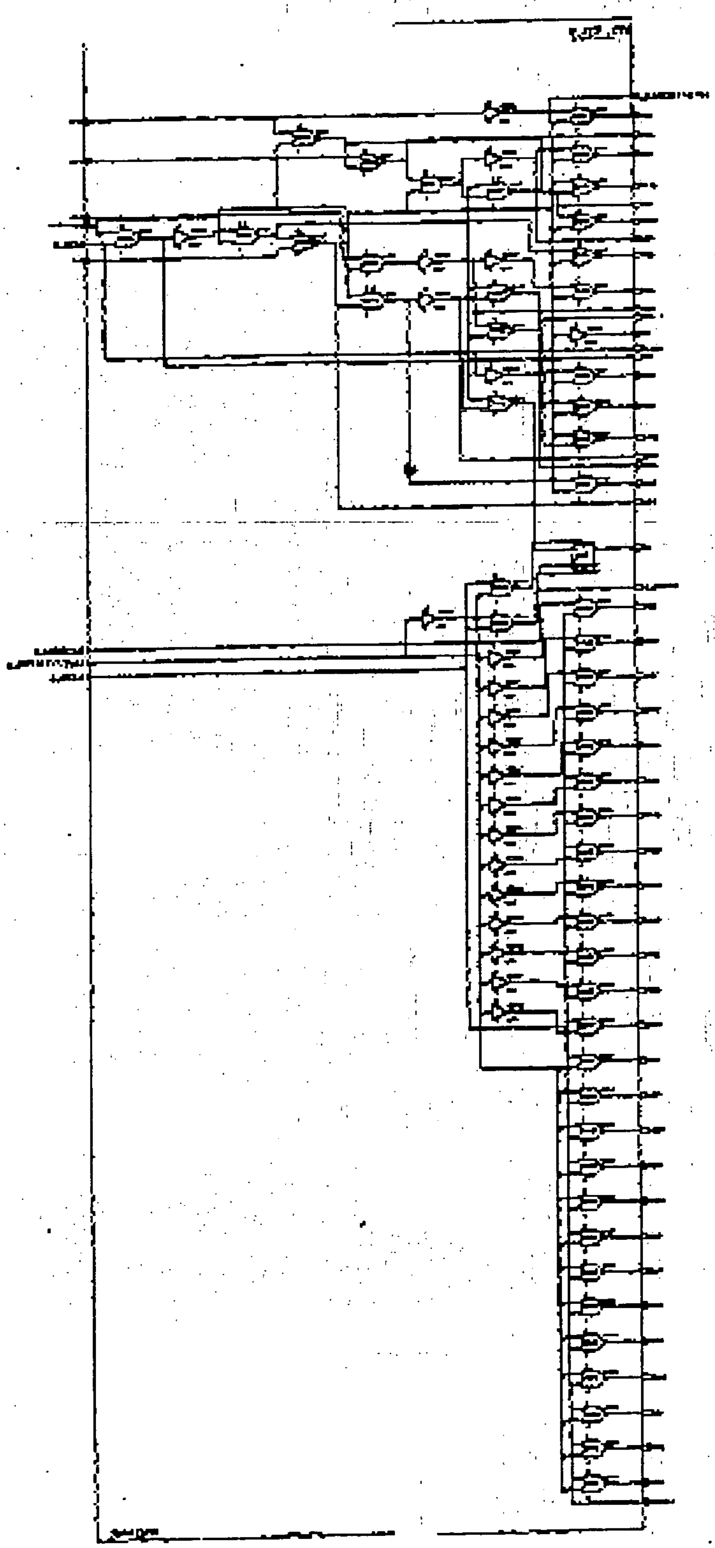


FIG. 57

TI 23L

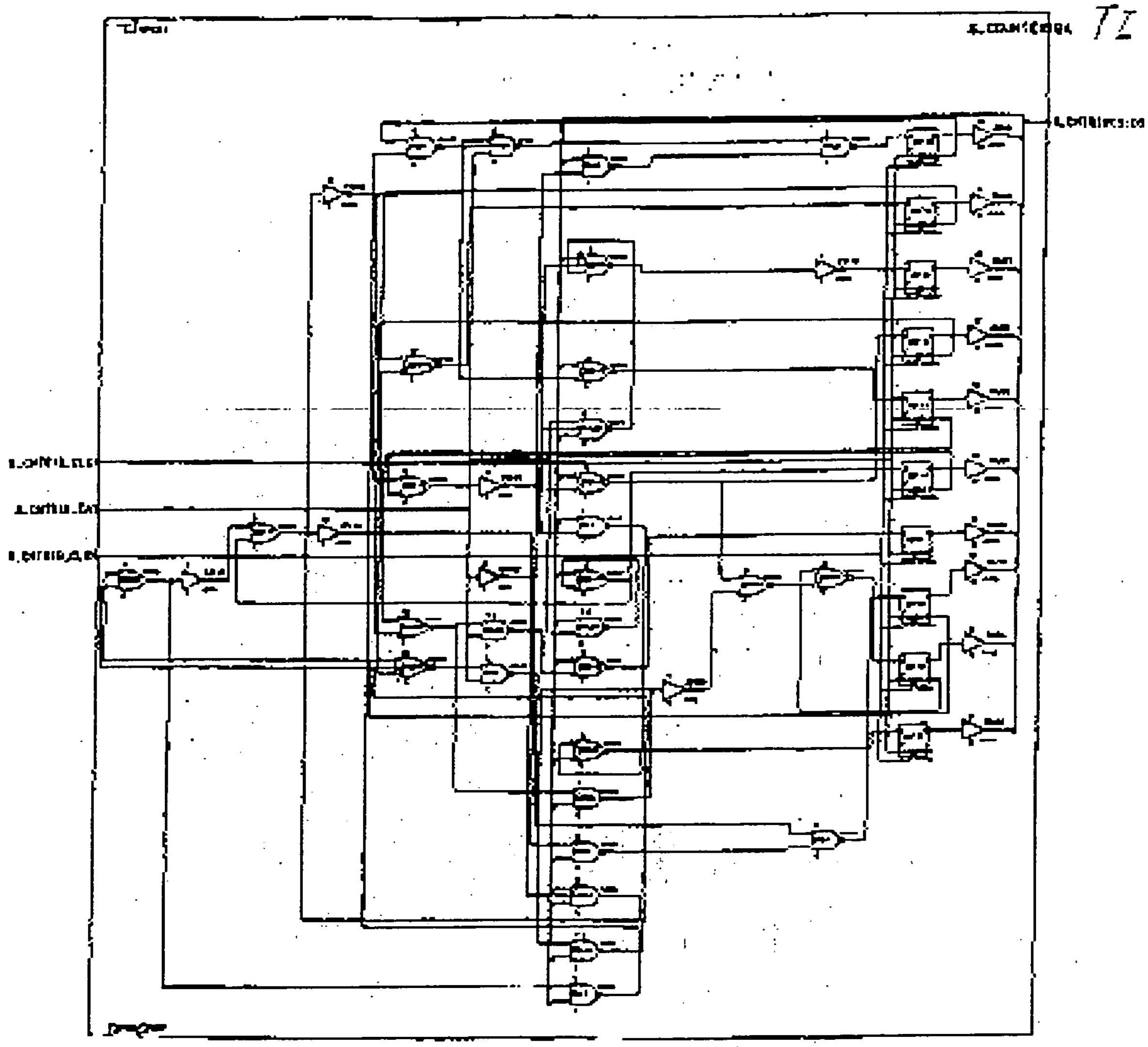


FIG. 58

TI 230

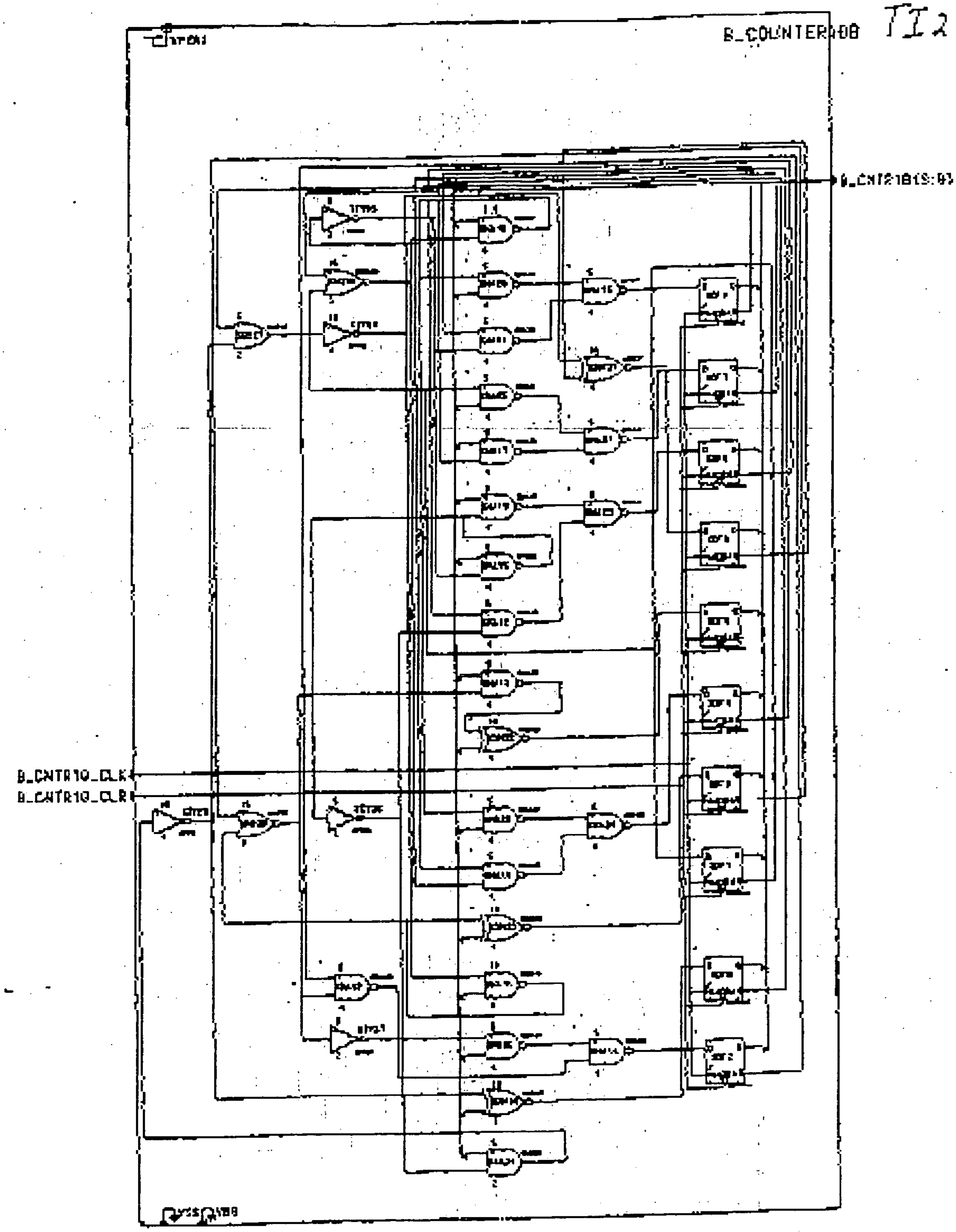


FIG. 59

Appendix: Figures and Diagrams

256Meg SDRAM BIST

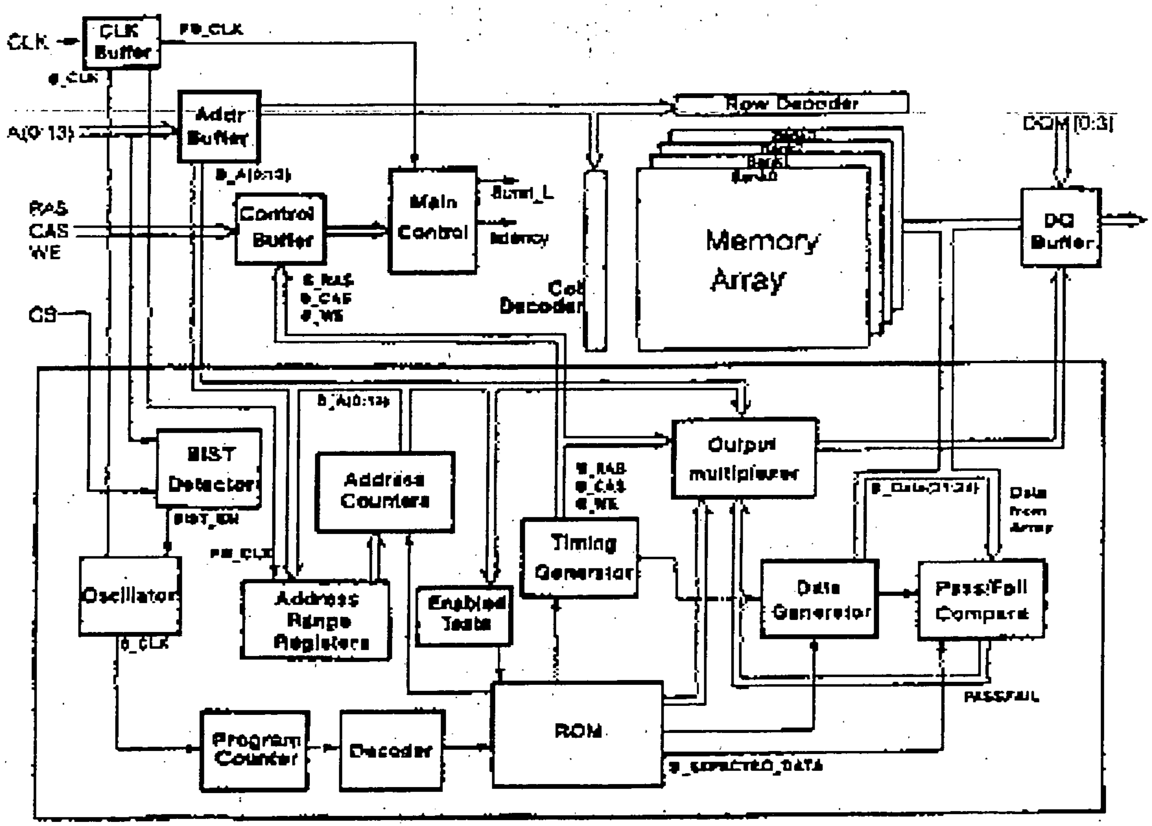


Fig 60

BIST block and Interface

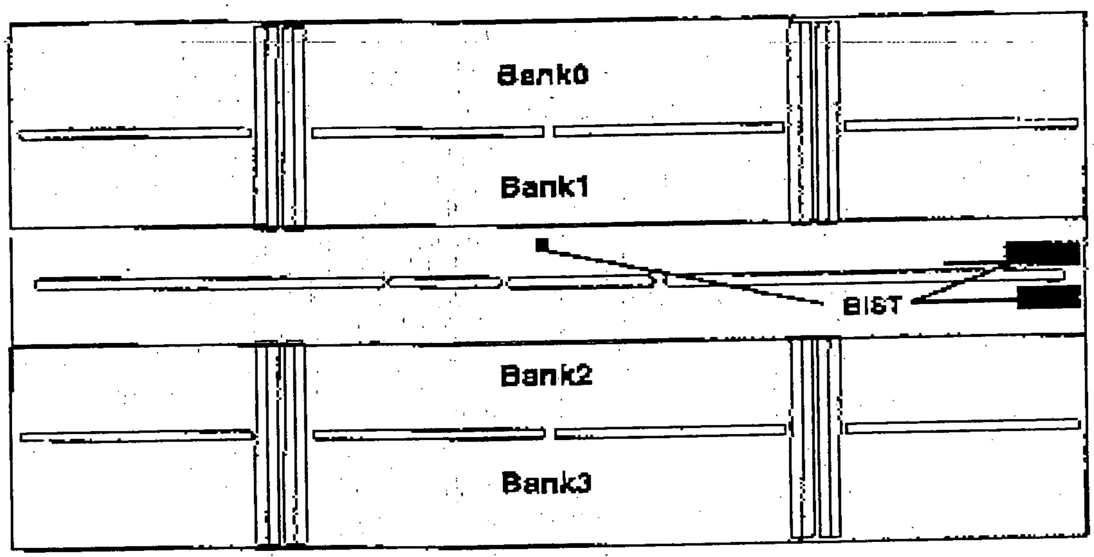


Fig 61

Algorithms	Timing Set	Addr Access	Array Access	Data Pattern	Misc option
Gross	Short	X, inc, dec	R, W	0, 1	CKBD, ALT
Pause	Short	X, Inc	R, W	0, 1	CKBD, Pause
Xmarch	Long	X, Inc	RW	0, 1	
Ymarch	Short	Y, Inc	RW	0, 1	
SDIST	Short	X,Y, Inc	R, W	0, 1	Refresh
LDIST	Long	X,Y, Inc	R, W	0, 1	Refresh
PDIST	Page	Y, Inc	R, W	0, 1	Refresh
Burn In	Long	X,Y, Inc	W	0, 1	CKBD
Eng 1	Short	Y, Inc, 1 row	W	1	
Eng2	Short	X, Inc, 1 Col	R	0	

Short – 300ns period
Long – 10,000ns period

R – Read
W – Write

inc – Increment
dec – Decrement

Fig 62

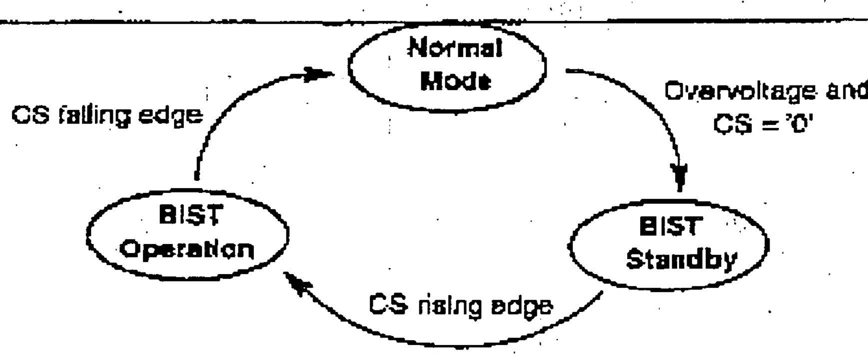


Fig 63

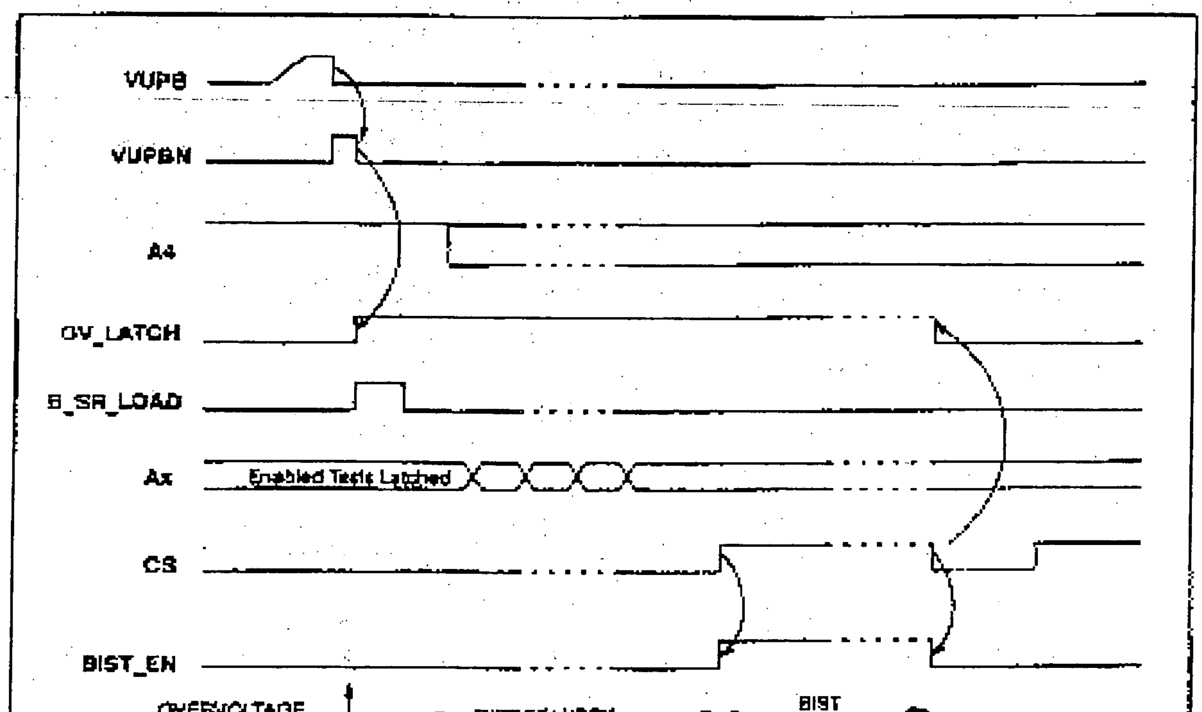

VUPB
VUPBN
A4
OV_LATCH
B_SR_LOAD
Ax
Enabled Tests Latched
CS
BIST_EN
OVERVOLTAGE DETECTION
BIST STANDBY
BIST OPERATION

Fig 64

256M SDRAM

3.3V	VDD			VSS	0.0V
No connect	DQ0	2	87	DQ15	No connect
3.3V	VDDQ	3	86	VSSQ	0.0V
No connect	DQ1	4	85	DQ14	No connect
No connect	DQ2	5	84	DQ13	No connect
0.0V	VSSQ	6	83	VDDQ	3.3V
No connect	DQ3	7	82	DQ12	No connect
No connect	DQ4	8	81	DQ11	No connect
3.3V	VDDQ	9	80	VSSQ	0.0V
No connect	DQ5	10	79	DQ10	No connect
No connect	DQ6	11	78	DQ9	No connect
0.0V	VSSQ	12	77	VDDQ	3.3V
No connect	DQ7	13	76	DQ8	No connect
0.0V	DQM0	14	75	DQM1	0.0V
3.3V	VDD	15	74	VSS	0.0V
No connect	NC	16	73	NC	No connect
3.3V	W	17	72	NC	No connect
3.3V	CE	18	71	Vref	1.65V
3.3V	RE	19	70	NC	No connect
0.0V / 3.3V	S	20	69	CLK	0.0V
0.0V / 3.3V	BA1	21	68	CKE	3.3V
0.0V / 3.3V	A11	22	67	BA0	0.0V / 3.3V
3.3V	VDD	23	66	VSS	0.0V
0.0V / 3.3V	A0	24	65	NC	No connect
0.0V / 3.3V	A1	25	64	A10	0.0V / 3.3V
0.0V / 3.3V	A2	26	63	A9	0.0V / 3.3V
0.0V / 3.3V	A3	27	62	A8	0.0V / 3.3V
0.0V / 5.4V	A4	28	61	A7	0.0V / 3.3V
0.0V / 3.3V	A5	29	60	A6	0.0V / 3.3V
3.3V	VDD	30	59	VSS	0.0V
0.0V	DQM2	31	58	DQM3	0.0V
DONE	DQ16	32	57	DQ31	No connect
0.0V	VSSQ	33	56	VDDQ	3.3V
PASSFAIL	DQ17	34	55	DQ30	No connect
No connect	DQ18	35	54	DQ29	No connect
3.3V	VDDQ	36	53	VSSQ	0.0V
No connect	DQ19	37	52	DQ28	No connect
No connect	DQ20	38	51	DQ27	No connect
0.0V	VSSQ	39	50	VDDQ	3.3V
No connect	DQ21	40	49	DQ26	No connect
No connect	DQ22	41	48	DQ25	No connect
3.3V	VDDQ	42	47	VSSQ	0.0V
No connect	DQ23	43	46	DQ24	No connect
3.3V	VDD			VSS	0.0V

Fig 65

	Normal Mode	BIST : Int Clk (B_CLKMUX = 0)	BIST : Ex Clk (B_CLKMUX = 1)
B_CLK	Low	20ns Clock	Same as Ext Clk
B_IRCLK	0.5Us Clock	Low	Low

Fig 68

256M BIST ALGO STORAGE CONFIGURATION

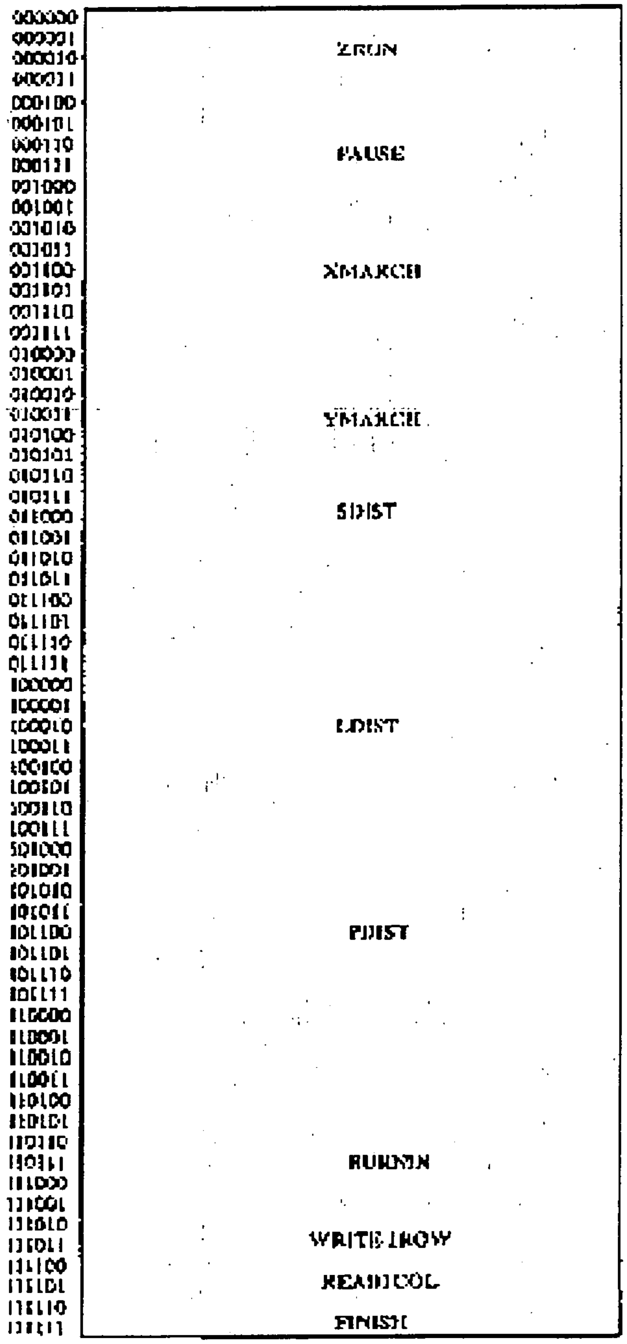


Fig 67

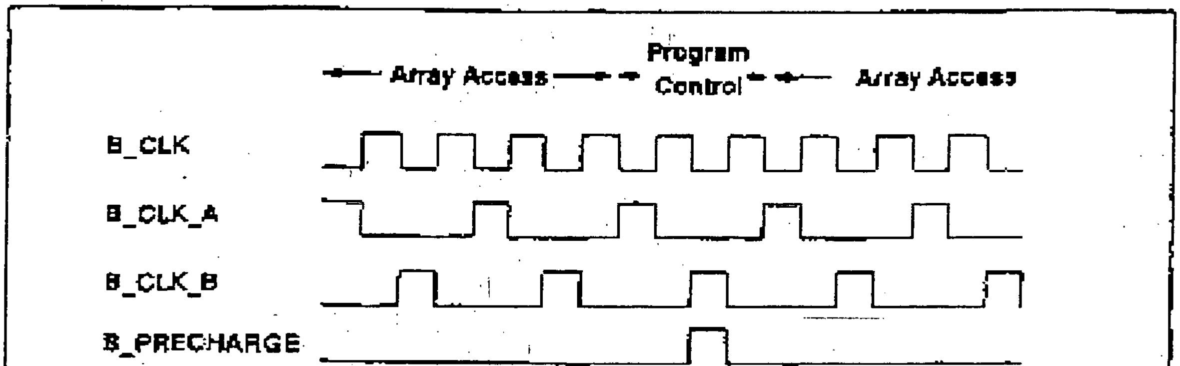


Fig 68

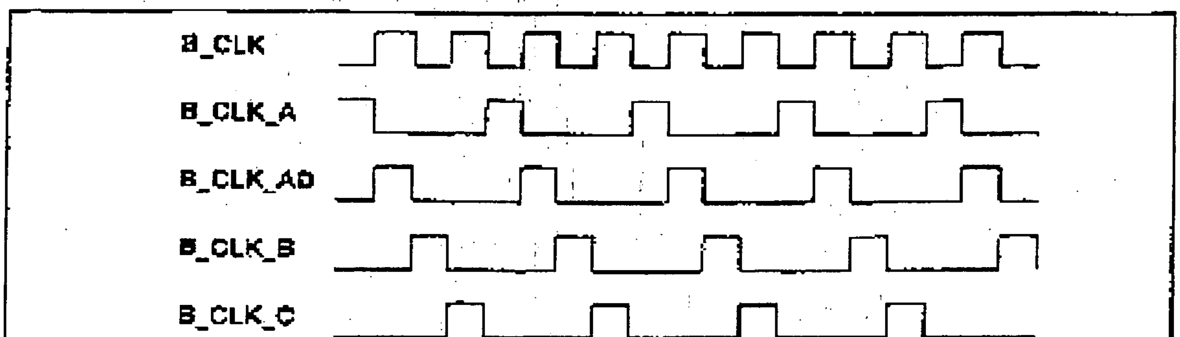


Fig 69

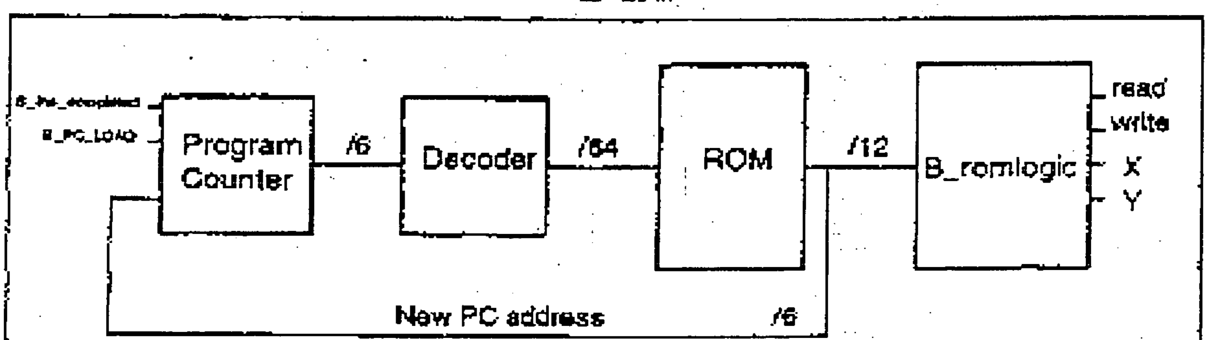


Fig 70

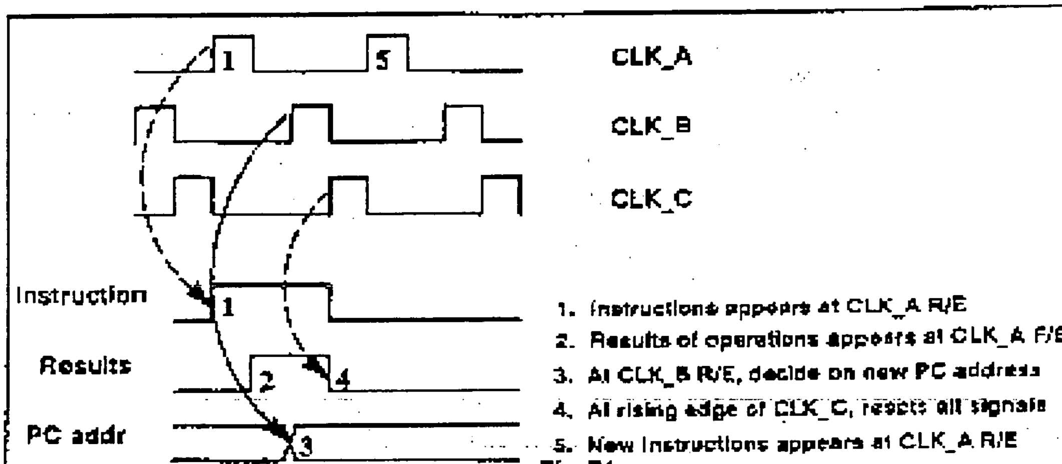
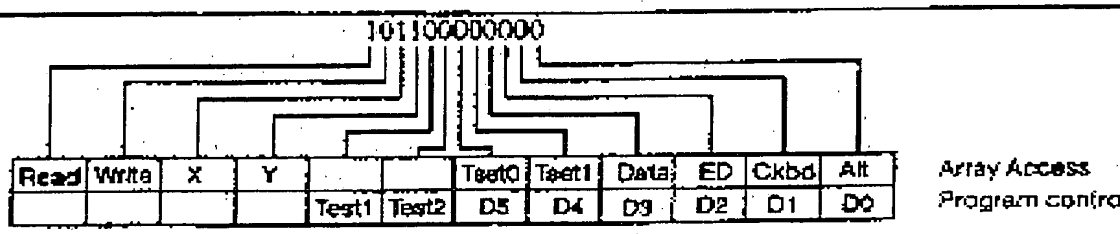


Fig 71



Test1	Test2	Comments
0	0	Test if the test is enabled
0	1	Test for inverted pattern
1	0	Increment address by 1, and check for overflow
1	1	Increment counter and check if required value is reached

Test0	Test1	Comments
0	0	Timing set A
0	1	Timing set C
1	0	Timing set B
1	1	Timing set A, decrement

Fig 72

Timing sets	Access mode	Addressing	Array size	'0' or '1'	Data Pattern
TSETA	READ	INC	FULLA	PATTERN0	CKBD
TSETB	WRITE	DEC	ROW	PATTERN1	ALTERNATE
TSETC	RMW		COL		

Fig 73

BIST Timing Generation

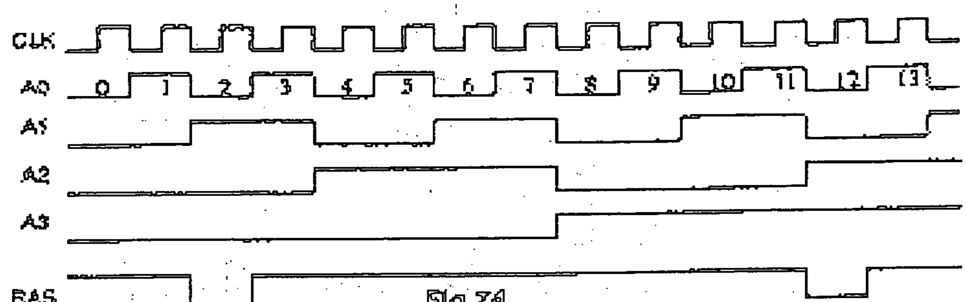
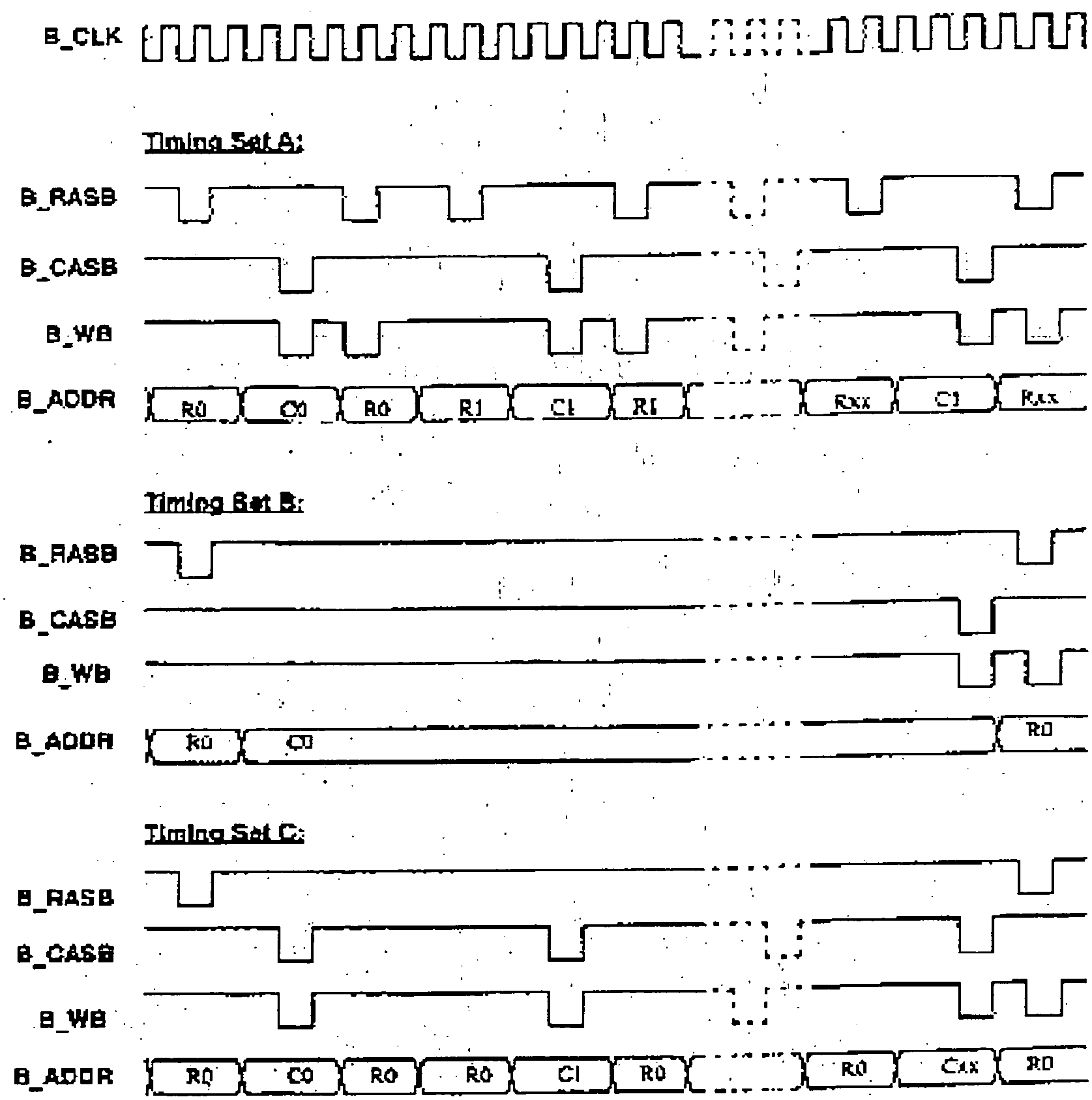


Fig 74

BIST Timing Sets



Timing Set	Cycle Time	Description
A	300ns	Short timing, used in Grass,Pause,Ymarch,SOIST,Eng1,Eng2
B	10,000ns	Long Timing, used in Xmarch, LDIST, Burn In
C	300ns	Multiple columns, used in PDIST

Fig 75

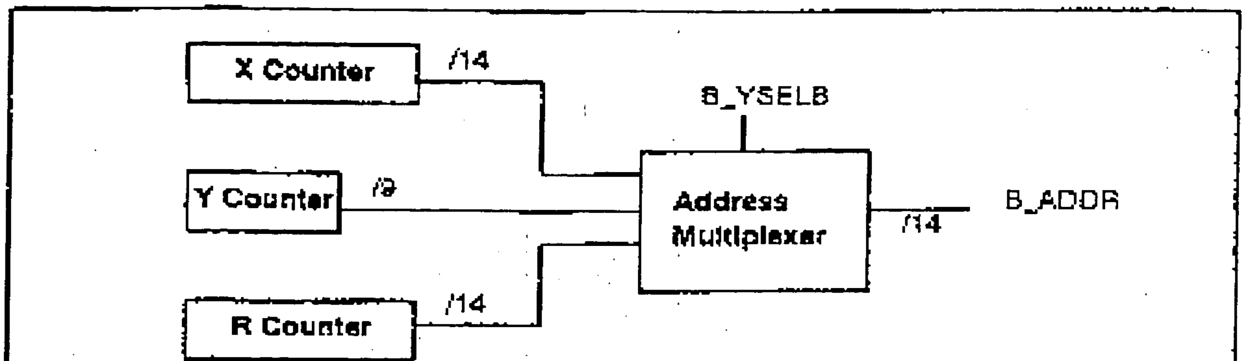


Fig 76

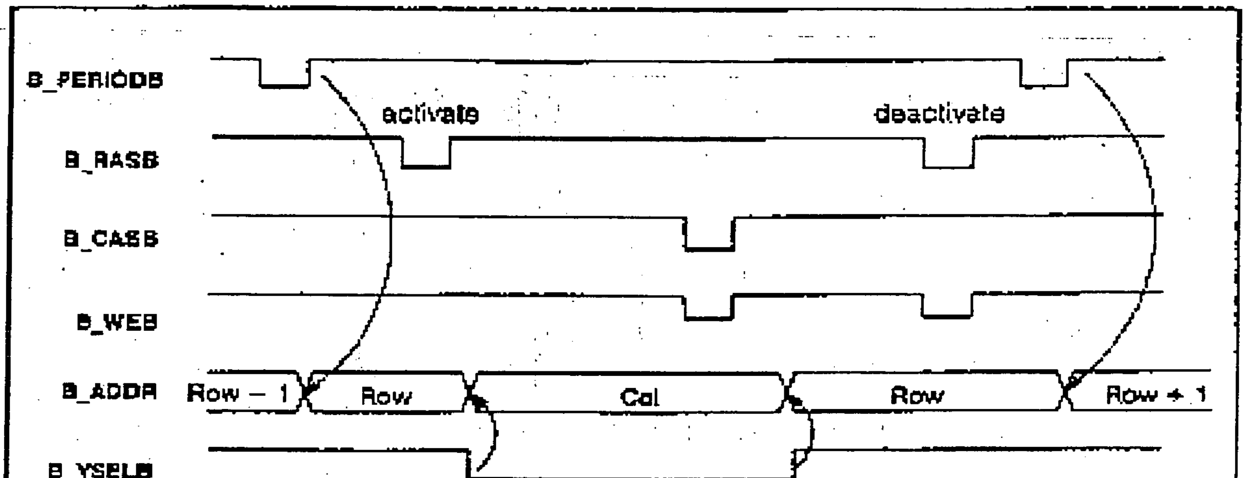


Fig 77

BA1	BA0	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Int/Ext CLK	Sub Array	Output Enable	R1C	W1R	BI	Pdlel	Ldlel	Sdlst	OV	Ymar	Xmar	Paues	Groes

For A0 to A3, '1' = Test of concerned enabled '0' = Test of concerned not enabled
For A4 = Overvoltage, 2 to 2 1/2 times VCC
For A5 to A10, '1' = Test of concerned enabled '0' = Test of concerned not enabled
For BA0, '1' indicate sub array selected, '0' indicate full array test.
For BA1, '1' indicate external clock. '0' indicate internal clock.
For A11, '1' indicate output enable, 0 indicate output not enabled. Only meaningful after BIST is done

Fig 78

	NORMAL	TLROMR	MON1	MON2	MON3
B_DQ(31)	High Z	RomBit 11	BA1	BA1	BA1
B_DQ(30)	High Z	RomBit 10	BA0	BA0	BA0
B_DQ(29)	High Z	RomBit 9	ADDR11	ADDR11	ADDR11
B_DQ(28)	High Z	RomBit 8	ADDR10	ADDR10	ADDR10
B_DQ(27)	High Z	RomBit 7	ADDR8	ADDR9	ADDR9
B_DQ(26)	High Z	RomBit 6	BIST_CMP	ADDR8	BIST_CMP ADDR8
B_DQ(25)	High Z	RomBit 5	B_DODD	ADDR7	B_DODD ADDR7
B_DQ(24)	High Z	RomBit 4	B_DEVEN	ADDR6	B_DEVEN ADDR6
B_DQ(23)	High Z	RomBit 3	B_RASB	ADDR5	B_RASB ADDR5
B_DQ(22)	High Z	RomBit 2	B_CASB	ADDR4	B_CASB ADDR4
B_DQ(21)	High Z	RomBit 1	B_WB	ADDR3	B_WB ADDR3
B_DQ(20)	High Z	RomBit 0	B_CLK	ADDR2	B_CLK ADDR2
B_DQ(19)	High Z	Low	ADDR1	ADDR1	ADDR1
B_DQ(18)	High Z	Low	ADDR0	ADDR0	ADDR0
B_DQ(17)	PASSFAIL	High Z	High Z	High Z	High Z
B_DQ(16)	B_DONE	High Z	High Z	High Z	High Z

Fig 79

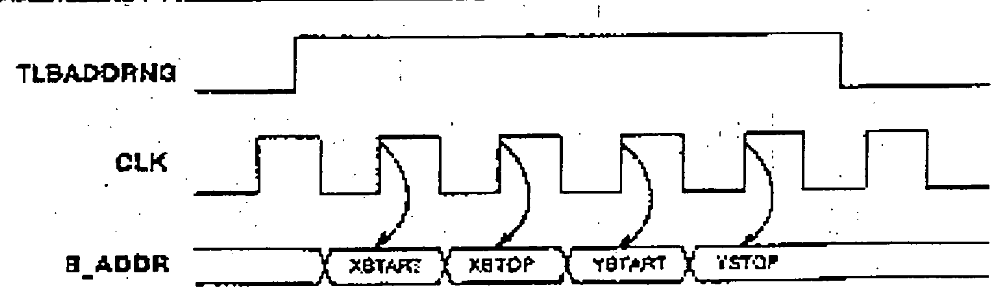


Fig 80

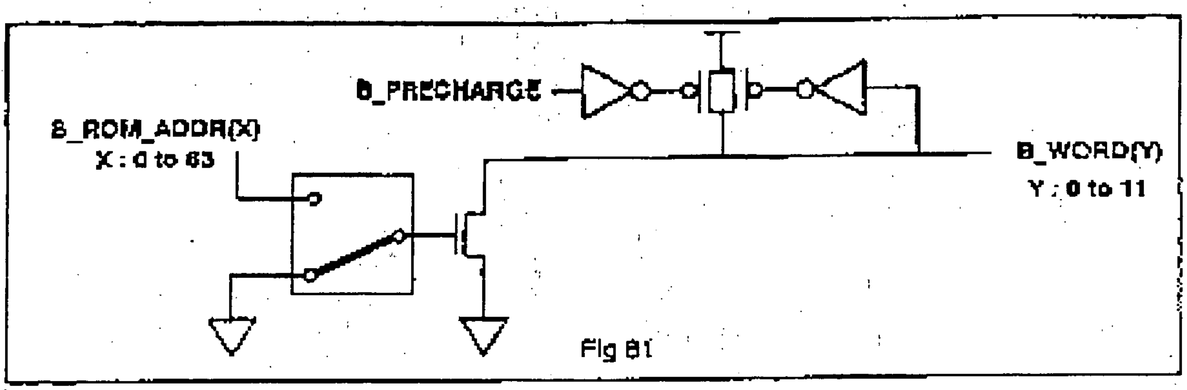


Fig 81

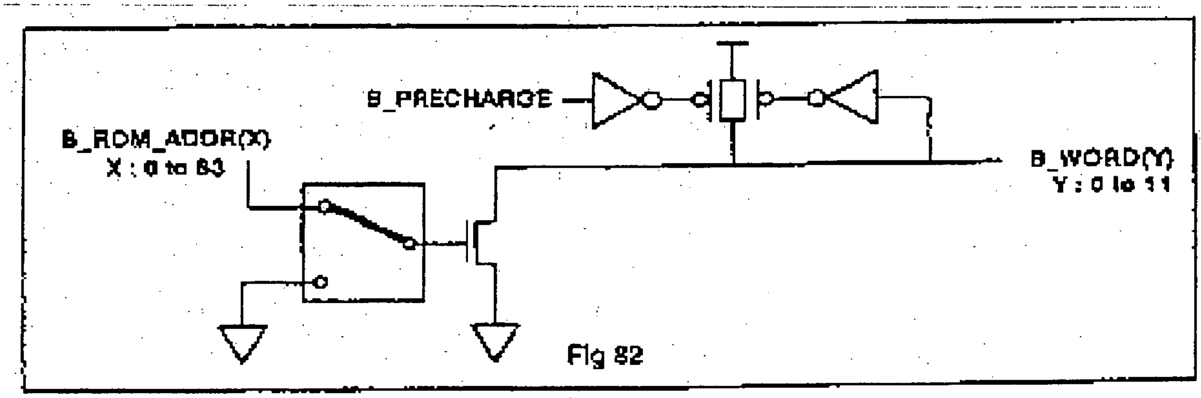


Fig 82

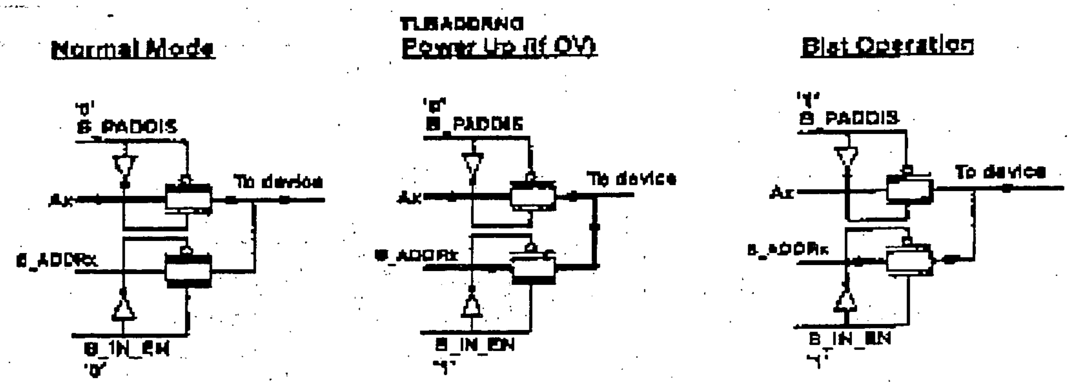


Fig 83